

POLITECNICO DI TORINO

Collegio di Ingegneria Informatica, del Cinema e Meccatronica

**Corso di Laurea Magistrale
in Ingegneria Informatica**

Tesi di Laurea Magistrale

La Business Intelligence in Rai Pubblicità Dashboard interattive e Chat Bot in Qlik Sense



Relatore
prof.ssa Elena Baralis

Candidato
Mario Amato

Anno Accademico 2017/2018

Ai miei genitori Laura e Saverio

Indice

Introduzione	1
1. Stato dell'arte	
1.1 Cos'è la Business Intelligence?	3
1.2 Strumenti e tecnologie a supporto della BI	6
1.3 Uso di NLP e Chat Bot in ambito BI	8
2. Software utilizzati	
2.1 SQL Developer	12
2.2 Qlik Sense	12
3.3 Google Dialogflow	14
3.4 Telegram	17
3. Sviluppo delle app "AQC" e "Flow"	
3.1 Specifiche e pianificazione del lavoro	19
3.2 Sistema aziendale di BI	21
3.3 Reperimento delle informazioni utili	25
3.4 Creazione delle viste predisposte per Qlik	27
3.5 Configurazione del framework ETL	36
3.5 Cenni sulla creazione del modello dati	37
3.6 Implementazione del "Front-End" delle app	43
4. Realizzazione di un Chat Bot Telegram	
4.1 Scelte progettuali	56
4.2 Architettura e implementazione del Chat Bot	57
4.3 Customizzazione del Chat Bot sulla base dell'app "FLOW"	61
Conclusioni	65
Bibliografia e sitografia	66
Ringraziamenti	68

Indice delle figure

Figura 1 - Piramide di Anthony	4
Figura 2 – Piramide DIKW (Data, Information, Knowledge, Wisdom)	5
Figura 3 - Esempi di ipercubo	7
Figura 4 - Modello strumenti di BI tradizionali vs Modello associativo Qlik	13
Figura 5 - Schema riassuntivo del funzionamento di Dialogflow	15
Figura 6 - Dialogflow Analytics	16
Figura 7 - Architettura di un generico Chat Bot Telegram	17
Figura 8 - Immagini estratte dal file "AQC - Dati per layout.xlsx"	20
Figura 9 - Architettura ETL aziendale "4 Tier QVD Based"	23
Figura 10 - Segregazione orizzontale e verticale dei dati nel sistema Rai Pubblicità	25
Figura 11 - Struttura del data model costruito per l'app "AQC"	38
Figura 12 - Struttura del data model costruito per l'app "Flow"	42
Figura 13 - Foglio "Selezione Cliente"	48
Figura 14 - Foglio "Scheda informativa"	49
Figura 15 - Foglio "Investimenti Rai Pubblicità"	50
Figura 16 - Foglio "Analisi mercato"	51
Figura 17 - Foglio "Focus TV/Radio", nessun target selezionato (in alto), target "Adulti 25-54" selezionato (in basso)	53
Figura 18 - Foglio "Focus web"	54
Figura 19 - Applicazione "Flow"	55
Figura 20 - Architettura del Chat Bot	57
Figura 21 - Creazione della entity "@Dimension"	62
Figura 22 - Creazione dell'intent "qs-ShowMeasureForElement"	63

Introduzione

Il presente elaborato è il risultato del lavoro da me svolto durante l'attività di tirocinio curriculare presso l'azienda Rai Pubblicità s.p.a. e si propone di illustrare, passo dopo passo, il mio operato e le numerose conoscenze che tale attività mi ha consentito di acquisire.

“Rai Pubblicità” è il nome che ha assunto, a partire dal 2013, la Sipra (Società Italiana per la Pubblicità Radiofonica Anonima), ossia una società per azioni, concessionaria del gruppo Rai, che gestisce in esclusiva la pubblicità su tutti i mezzi (radio, tv e web) e le piattaforme Rai. Tale società ha inoltre in gestione la pubblicità dei principali circuiti cinematografici in Italia e si è contraddistinta, negli ultimi anni, per l'innovazione e la sperimentazione di nuovi format pubblicitari.

Il mio ingresso all'interno dell'azienda è avvenuto nel mese di giugno 2017 e l'attività di stage si è protratta fino ad aprile 2018, sotto l'attenta supervisione di Mauro Sandri, responsabile applicazioni interfunzionali e business intelligence, nonché mio tutor aziendale. Nello specifico, la mia figura è stata inserita all'interno del team aziendale di Business Intelligence, il cui scopo è quello di fornire ai “business users” degli strumenti innovativi che consentano un accesso dinamico ed intuitivo alle informazioni utili, rendendo molto più semplici ed immediati i processi decisionali. Tali strumenti si propongono di sostituire l'obsoleto modello del report cartaceo, che vincola l'utente ad una consultazione statica dei dati.

In tale contesto il mio lavoro ha riguardato la progettazione e lo sviluppo di un'applicazione finalizzata all'analisi degli investimenti dei principali clienti di Rai Pubblicità, che fosse in grado di fornire supporto sia ai top manager, nel prendere decisioni, sia agli addetti alla vendita nello svolgere i loro compiti. L'intera attività è stata condotta in collaborazione con il mio amico e collega Enrico Manenti, il quale si è soffermato prevalentemente sugli aspetti inerenti alla progettazione e alla realizzazione del modello dati sottostante all'applicazione, trattati in maniera esaustiva all'interno dello scritto “La Business Intelligence in Rai Pubblicità, Studio del framework di ETL e realizzazione del data model Qlik nell'ambito di un caso reale”, ed è per questo motivo che, all'interno della presente tesi, tali argomenti, pur facendo parte del lavoro svolto, non verranno trattati in modo approfondito. Verrà trattato in modo minuzioso, invece, l'insieme delle tecniche utilizzate relativamente alla presentazione delle informazioni e alla fruizione di queste ultime da parte dell'utente finale.

Di seguito verrà tracciata, innanzitutto, una panoramica della Business Intelligence, ponendo particolare attenzione agli strumenti di supporto che è possibile utilizzare in tale ambito; dopodiché verrà illustrato l'iter seguito al fine di produrre le applicazioni “Accordo Quadro Cliente” e “Flow”, realizzate su piattaforma Qlik Sense, indicando quelli che sono stati i software utilizzati, spiegando il funzionamento del framework di caricamento dati aziendale e descrivendo in maniera dettagliata in che modo le suddette applicazioni siano state implementate. Infine, verrà posta l'attenzione su un lavoro aggiuntivo da me svolto con l'obiettivo di realizzare un “Chat Bot” aziendale che, supportato da un sistema di processamento del linguaggio naturale, risulta in grado di interfacciarsi con le applicazioni realizzate precedentemente e di estrarre informazioni da tali applicazioni, per rispondere alle richieste che gli utenti formulano, utilizzando un linguaggio colloquiale.

Per motivi di privacy, i dati esposti nel presente elaborato, che fanno riferimento ai clienti di Rai Pubblicità e agli investimenti effettuati da questi ultimi, in quanto informazioni riservate e non divulgabili, non coincidono con i dati realmente presenti all'interno del sistema informativo aziendale, ma ne costituiscono una versione alterata.

1. Stato dell'arte

In questo capitolo verrà effettuata una trattazione approfondita della Business Intelligence (BI), spiegando l'etimologia del termine "Business Intelligence", l'insieme delle attività racchiuse sotto questo nome e i motivi che hanno spinto e spingono moltissime aziende ad adottare tali tecniche. Successivamente verranno illustrati alcuni degli strumenti ideati al fine di supportare la Business Intelligence, i quali consentono una semplificazione del lavoro di coloro che operano in tale settore. Per concludere, poi, sarà tracciata una panoramica relativa a "Natural Language Processing" e "Chat Bot" e, in particolare, verrà spiegato come tali tecnologie possano essere utilizzate nell'ambito della BI.

1.1 Cos'è la Business Intelligence?

Il termine "Business Intelligence" venne utilizzato per la prima volta da Hans Peter Luhn, ricercatore ed inventore tedesco che, nell'articolo "A Business Intelligence System", pubblicato dall'IBM Journal nel 1958, ipotizzava l'utilizzo di un sistema automatico di diffusione delle informazioni all'interno di un contesto aziendale. La paternità del concetto di Business Intelligence, però, non viene attribuita a Luhn, in quanto quest'ultimo coniò il termine in un periodo in cui le operazioni di business non erano ancora computerizzate e la BI, nonostante assolvga oggi ad alcuni dei propositi indicati all'interno dell'articolo, si è sviluppata in una direzione che lo stesso Luhn non avrebbe mai potuto prevedere. La mancanza di una piena attinenza tra la definizione data da Luhn ed il significato che ha al giorno d'oggi il termine Business Intelligence, dunque, portano a riconoscere come ideatore della BI Howard Dresner, analista di Gartner Group, che nel 1989 asserì: "La Business Intelligence è la capacità di un'impresa di accedere e navigare le informazioni contenute all'interno di un Datawarehouse e di analizzare tali informazioni con l'obiettivo di sviluppare approfondimenti e comprensione, che portino ad un processo decisionale migliorato ed informato"¹. Naturalmente, quanto affermato da Dresner definisce in maniera molto semplificata un concetto che, come vedremo nel seguito di questo capitolo, risulta molto più complesso ed articolato.

Per meglio comprendere cosa sia la Business Intelligence, risulta necessaria una digressione sui sistemi informativi aziendali, poiché questi costituiscono le fondamenta a partire dalle quali si sviluppa tutta la BI. Il sistema informativo aziendale è definito come l'insieme delle attività, dei processi, delle risorse tecnologiche e delle persone che sono preposte alla gestione delle informazioni all'interno di un'azienda ed ogni operazione svolta dal sistema informativo può essere classificata secondo la cosiddetta "Piramide di Anthony", ossia una struttura piramidale all'interno della quale è possibile distinguere tre diverse categorie di attività aziendali:

- Al vertice della Piramide di Anthony sono collocate le attività strategiche, volte a definire gli obiettivi dell'azienda e le politiche che l'azienda dovrebbe adottare al fine di raggiungerli.

¹ Fonte: www.gartner.com

- Al livello sottostante si trovano le attività tattiche che hanno come finalità primaria quella di applicare le direttive ricevute dal livello superiore e di monitorare il lavoro svolto a livello più basso.
- Alla base della piramide, infine, troviamo le attività operative, ossia le attività effettivamente svolte dall'azienda.

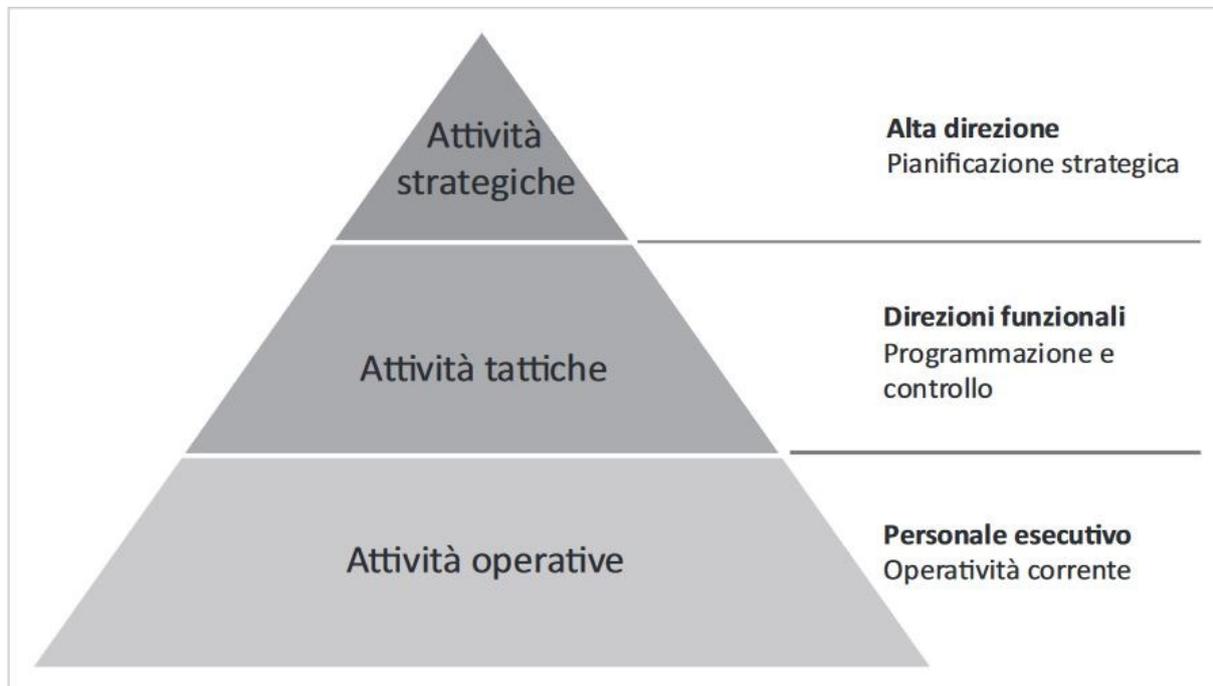


Figura 1 - Piramide di Anthony

Secondo Henry Mintzberg, studioso di scienze gestionali, ricerca operativa, organizzazione e strategia, le figure aziendali con un ruolo manageriale hanno sempre meno tempo da impiegare nello studio della propria attività e da ciò discende un loro approccio al lavoro disorganizzato e poco efficiente. Questo problema, è generalmente riscontrabile a tutti i livelli della piramide di Anthony, ma risulta vitale per le aziende risolverlo laddove si svolgono le attività strategiche. A questo punto entra in gioco la BI che evita alle persone, specialmente a coloro che prendono le decisioni, di preoccuparsi delle fasi di reperimento, elaborazione e conservazione dei dati, consentendogli un'interazione migliore con i dati stessi. Lo scopo della BI, come sottolineato anche dalla spiegazione fornita da Dresner, è quindi quello di aiutare i manager ad analizzare le informazioni riguardanti il proprio settore, rapidamente, senza che essi sprechino tempo a reperirle. Appare chiaro, dunque, che per un'azienda, prima di mettere in piedi un sistema di Business Intelligence, è fondamentale dotarsi di un sistema informativo ben progettato.

Un sistema di BI può essere assimilato ad una piramide, alla cui base si trovano i dati grezzi, spesso destrutturati o frammentati, provenienti da fonti eterogenee tra di loro (file, documenti, database, etc.). Risalendo la piramide si ottengono dati sempre più distillati, fino a raggiungere la sintesi estrema, ossia un insieme di informazioni essenziali, presentate in modo semplice, che possano costituire effettivamente un supporto alle decisioni. Di seguito vengono analizzati tutti gli step da seguire per percorrere la piramide dalla base al vertice, ottenere la trasformazione dei dati in informazioni e, successivamente, delle informazioni in conoscenza:

- Innanzitutto, viene effettuata la raccolta dei dati che, come già detto, sono disorganizzati e provenienti da sorgenti diversificate. I dati raccolti costituiscono la base della piramide, ma non hanno alcun significato, in quanto non sono calati all'interno di un contesto. Per capire meglio l'importanza della contestualizzazione di un dato, è possibile considerare come esempio la stringa 01012018: tale stringa è apparentemente una sequenza numerica senza importanza, ma se etichettata come “data” assume un significato ed un'importanza completamente diversi. Attraverso le fasi di pulizia, validazione ed integrazione, i dati raccolti vengono scremati, eliminando eventuali errori o duplicati e vengono memorizzati all'interno di strutture dati aziendali, quali datawarehouse o data mart.
- Il passo successivo consiste nel trasformare i dati in informazioni, contestualizzandoli. Questo è reso possibile grazie ad un processo di elaborazione, aggregazione ed analisi. Data l'ingente mole di dati, le informazioni rilevanti vengono individuate in quanto rispondono alle domande “chi?”, “cosa?”, “quando?” e “dove?”.
- Trasformato il dato in informazione, dunque, si effettua un'ulteriore analisi che consente la trasformazione dell'informazione in conoscenza. Questo risultato si ottiene nel momento in cui si riesce a rispondere alla domanda “in che modo questa informazione è utile al raggiungimento degli obiettivi aziendali?”; trovando la risposta a questa domanda, infatti, l'informazione non viene più vista come mera descrizione di un fatto, ma come mezzo per il raggiungimento di un fine.
- L'ultimo step prevede la trasformazione della conoscenza in saggezza, nonché il raggiungimento della punta della piramide, e viene compiuto nel momento in cui, sulla base dei risultati ottenuti ai livelli sottostanti, viene presa una decisione e dati, informazioni e conoscenza si tramutano in azione.

Si potrebbe affermare che i dati raccolti e le informazioni ricavate siano entità relative al passato, mentre la conoscenza e la saggezza che ne derivano appartengano al presente e facciano riferimento ad un obiettivo che si prevede di raggiungere in futuro.



Figura 2 – Piramide DIKW (Data, Information, Knowledge, Wisdom)

1.2 Strumenti e tecnologie a supporto della BI

Fino ad ora si è guardato alla Business Intelligence intesa come processo aziendale, ma quando si parla di BI, inevitabilmente, si fa riferimento anche a tutte le tecnologie e a tutti gli strumenti utilizzati al fine di supportare tale processo. Sarebbe impossibile eseguire le fasi di raccolta, elaborazione e validazione manualmente, vista la mole enorme di dati collezionati da alcune aziende ed è per questo motivo che tutte le operazioni di manipolazione su dati ed informazioni sono inserite all'interno di processi aziendali automatizzati e supportate da appositi strumenti software. In questo paragrafo si è scelto di focalizzare l'attenzione solamente su quella parte di prodotti più diffusa ed utilizzata.

- ETL (Extraction, Transform, Load): è uno strumento che consente l'estrazione dei dati da sorgenti diversificate (database transazionali, file di testo, etc.) per caricarli all'interno di un sistema di sintesi (un datawarehouse o un data mart), facendo in modo che l'insieme di dati finale risulti omogeneo e utile alle finalità dell'azienda. Una volta estratti, i dati subiscono un processo di trasformazione che ha come obiettivi la selezione dei soli campi di interesse, la normalizzazione dei dati, la traduzione dei dati codificati, la derivazione di nuovi campi calcolati a partire da quelli esistenti, la combinazione di dati provenienti da sorgenti differenti ed il loro raggruppamento. Il caricamento può effettuare un "refresh", ossia ricaricare per intero i dati, oppure un "update", che consiste nel caricare solamente i dati che hanno subito variazioni rispetto all'ultimo caricamento.
- Datawarehouse: letteralmente significa "magazzino di dati" e può essere visto come un archivio informatico che ospita i dati aziendali, in modo tale che questi siano facilmente fruibili per scopi decisionali. A differenza dei normali database relazionali, l'obiettivo del datawarehouse non è quello di normalizzare i dati, bensì quello di denormalizzarli, in modo da ottenere una ridondanza che ottimizza i tempi di risposta alle interrogazioni e fornisce una visione multidimensionale delle informazioni. Peculiarità del datawarehouse sono la possibilità di integrare dati provenienti da fonti eterogenee, il fatto che ogni datawarehouse sia dedicato al supporto di una specifica area aziendale, la capacità di memorizzare informazioni relative ad un particolare momento storico e la non volatilità dei dati, ottenuta grazie alla possibilità di accedervi solo in lettura.
- Data Mart: è un archivio alimentato dal datawarehouse e, di conseguenza, costituisce un sottoinsieme logico o fisico di quest'ultimo. La differenza tra datawarehouse e data mart risiede nel fatto che, mentre il primo viene creato all'interno di un contesto generico, per poi essere adattato alle esigenze di un ambito specifico, il secondo viene creato già al fine di agevolare il lavoro di un determinato settore aziendale.
- OLAP (On Line Analytical Processing): è una delle tecnologie utilizzate per velocizzare il processo di interrogazione di una base dati e consiste nell'effettuare un pre-calcolo di tutte le possibili risposte alle interrogazioni inerenti ad un dato. I sistemi OLAP si dividono in MOLAP (Multidimensional On Line Analytical Processing) e ROLAP (Relational On Line Analytical Processing); i primi prendono le informazioni da un datawarehouse e le trasformano in dati multidimensionali, dando vita a quello che viene chiamato "cubo dimensionale" o "ipercubo", mentre i secondi prelevano le informazioni da un database relazionale. Esiste anche una terza soluzione, detta HOLAP (Hybrid On Line Analytical Processing), che consente di combinare gli approcci MOLAP e ROLAP.
- Sistemi di reportistica: sono sistemi montati al di sopra del datawarehouse preposti alla produzione di report, ossia documenti composti da tabelle e grafici che presentano le misure di rilievo per i fenomeni aziendali analizzati, destrutturate e disaggregate a seconda delle esigenze. Tali misure saranno utili, in seguito, per effettuare ulteriori analisi.

- Business Analytics: i due principali limiti della reportistica e della tecnologia OLAP risiedono nell'impossibilità di effettuare previsioni su possibili eventi futuri e nell'incapacità di individuare schemi non previsti tra i dati. Gli strumenti di business analytics, invece, attraverso l'utilizzo di tecniche di data mining e la creazione di modelli che simulano i processi aziendali, permettono il superamento dell'analisi retrospettiva ed introducono la possibilità di "identificare schemi di comportamento significativi e correlazioni tra le variabili entro un complesso insieme di dati, strutturati e non strutturati, storici, attuali e potenziali, allo scopo di prevedere eventi futuri e valutare le opportunità offerte da diverse linee d'azione"². In altre parole, il lavoro svolto per estrarre la conoscenza dalle informazioni, eseguito da apposite figure aziendali che sfruttano le proprie capacità di intuizione e deduzione, con i business analytics potrà essere svolto in maniera automatica ed oggettiva.

1.2.1 Rappresentazione multidimensionale dei dati

Prima di concludere la trattazione degli strumenti e delle tecnologie utilizzati come supporto al processo di Business Intelligence, risulta opportuno approfondire il concetto di rappresentazione multidimensionale dei dati. Ogni evento che si verifica inerente al mondo aziendale prende il nome di "fatto" e le possibili descrizioni quantitative dei fatti vengono dette "misure"; gli eventi, inoltre, possono essere analizzati da prospettive differenti e tali prospettive sono indicate come "dimensioni". Detto ciò, è possibile immaginare il fatto come un n-cubo i cui spigoli costituiscono le dimensioni di analisi e le cui celle corrispondono ad un valore di misura.

Nell'immagine sottostante, a sinistra, è rappresentato un cubo, all'interno del quale ogni cella rappresenta l'acquisto di un prodotto, effettuato da un cliente, in una certa data. A destra, invece, sono rappresentate le operazioni di "slicing" e "dicing" che consentono di selezionare un sottoinsieme dei dati presenti nel cubo, fissando una delle dimensioni possibili (slicing) o più di una delle dimensioni possibili (dicing).

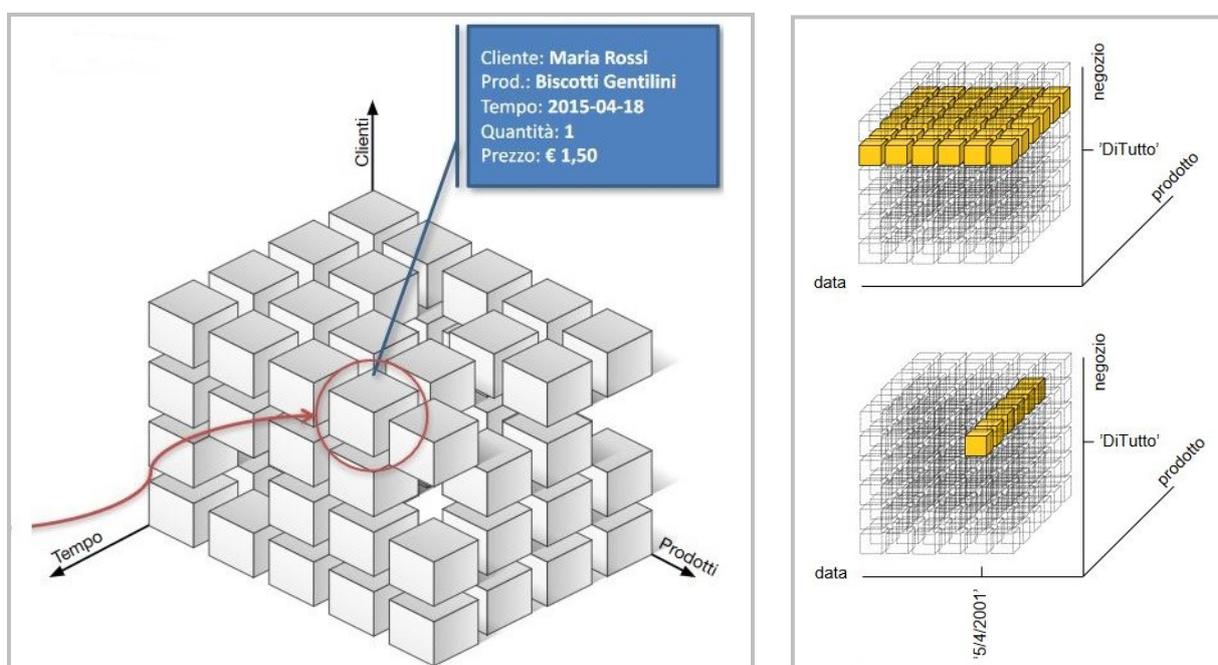


Figura 3 - Esempi di ipercubo

² www.forrester.com

1.3 Uso di NLP e Chat Bot in ambito BI

L'interazione uomo-macchina si è sempre basata sui linguaggi di programmazione, facilmente comprensibili dai computer, ma difficili da apprendere per l'uomo. Negli ultimi anni, tuttavia, grazie alla continua evoluzione delle scienze informatiche, si è reso possibile il dialogo tra persone e calcolatori mediante l'utilizzo del linguaggio umano, sebbene in una forma ancora acerba. L'esempio più evidente di suddetta "rivoluzione" è rappresentato da quelli che chiamiamo "assistenti vocali" (Siri, Google Now, Cortana, Amazon Alexa, etc.), i quali riescono a comprendere le richieste formulate dagli utenti e a rispondere a tali richieste in una forma comprensibile agli utenti stessi.

Nonostante le tecnologie a supporto di queste nuove forme di comunicazione siano ancora agli albori, molte ricerche hanno dimostrato che nei prossimi anni sempre più persone preferiranno utilizzare chat bot ed assistenti vocali, piuttosto che app e siti web, considerati poco pratici e meno immediati. La possibilità di richiedere le informazioni dialogando come se ci si stesse rivolgendo ad un amico, dunque, avrà come principale conseguenza quella di rendere obsoleto l'utilizzo delle interfacce utente per l'interazione con i computer.

Tutto ciò non è passato inosservato agli occhi di chi si occupa di Business Intelligence, poiché, essendo obiettivo della BI, come già detto, quello di migliorare la fruizione delle informazioni da parte di chi prende le decisioni all'interno dell'azienda, il poter formulare delle query, in qualunque momento e ovunque ci si trovi, semplicemente servendosi dei comandi testuali e vocali, disponibili su qualsiasi smartphone, costituisce un notevole miglioramento. Non saranno più necessari anni di esperienza e familiarità con gli strumenti software per comprendere come interrogare i sistemi aziendali al fine di estrarre i dati, ma basterà avere a disposizione un chat bot cui rivolgere tutte le domande.

1.3.1 Cos'è un Chat Bot?

Il termine "Chat Bot" fu coniato da Michael Mauldin per descrivere dei software progettati al fine di sostenere una conversazione con uno o più esseri umani. Un chat bot simula il comportamento umano e spesso si serve di un sistema di "Natural Language Processing" per comprendere e formulare frasi in linguaggio naturale.

Uno dei primi a pensare che i computer potessero conversare al pari delle persone fu Alan Turing, il quale, nel 1950, ideò un criterio, conosciuto oggi come "test di Turing", per determinare se una macchina fosse in grado di "pensare" o meno. Tale criterio viene soddisfatto ogniqualvolta un software, rispondendo a precise domande, riesca a indurre gli interlocutori a credere di parlare con un umano. La prima implementazione di chat bot si ebbe alcuni anni dopo, nel 1966, quando Joseph Weizenbaum progettò ELIZA, un software in grado di riconoscere parole o frasi ricevute come input e di produrre risposte tali da far progredire la conversazione, facendo credere che il computer abbia compreso. Ad ELIZA seguirono altri chat bot, quali A.L.I.C.E. e Jabberwacky, che aggiungevano alla possibilità di simulare conversazioni testuali, quella di effettuare ricerche sul web. Nel 2000 Robert Hoffer e Timothy Kay fondarono ActiveBuddy, una società votata alla creazione di intelligenze artificiali che fossero in grado di comunicare attraverso applicazioni di messaggistica istantanea. Tali entità intelligenti consentivano di aggiungere funzionalità ai servizi di messaggistica, anche interagendo con database e web services. Nel periodo recente sono numerosissime le aziende che hanno deciso di investire risorse nella realizzazione di agenti virtuali: si pensi ad esempio a Telegram che, dal 2015, offre supporto alla creazione di chat bot programmabili, oppure a Facebook che dal 2016 ha deciso di consentirne l'utilizzo sulla piattaforma Messenger.

Dalla comparsa del primo chat bot ad oggi, sono stati compiuti notevoli passi in avanti e ciò è dimostrato dal fatto che, nel 2014, un software ideato da Vladimir Veselov e Eugene Demchenko è riuscito, per la prima volta nella storia, a superare il test di Turing. Nonostante ciò, la capacità dei computer di colloquiare in modo lineare e congruo resta limitata, in quanto essi cercano di riprodurre i processi di apprendimento, comprensione e formulazione del linguaggio propri della mente umana, pur non possedendo molte doti che sono innate nell'uomo. Gli schemi che si trovano alla base del processo linguistico sono comunque oggetto di studi continui, che coinvolgono varie discipline, ed ogni nuova scoperta effettuata in tale ambito apporterà miglioramenti anche all'abilità di dialogare delle intelligenze artificiali.

Oggi giorno i chat bot vengono utilizzati principalmente per l'assistenza online, in sostituzione ai call-center, ma si prevede che, entro il 2025, il loro mercato arrivi a toccare 1,23 miliardi di dollari, con una crescita annua del 24,3%.

1.3.2 *Il Natural Language Processing*

I chat bot possono limitarsi a ricercare, all'interno delle frasi ricevute come input, delle parole chiave e a formulare delle risposte in base a queste, oppure, possono fare uso di un sistema per l'elaborazione del linguaggio naturale, anche detta "Natural Language Processing".

Il "Natural Language Processing" (NLP) non è altro che un connubio di intelligenza artificiale e linguistica e si pone l'obiettivo di rendere comprensibili ai computer le frasi o le parole scritte in linguaggio umano. Le componenti principali del NLP sono "Natural Language Understanding" e "Natural Language Generation" che servono, rispettivamente, alla comprensione e alla generazione del testo. Il Natural Language Understanding coincide con la linguistica ed ingloba le sei branche descritte di seguito.

- **Fonologia:** la parola "fonologia" deriva dai termini φωνή (*phōnē*, "voce", "suono") e λόγος (*lógos*, "parola", "discorso"), letteralmente "studio del suono", e serve ad indicare lo studio delle parole dal punto di vista del suono emesso per pronunciarle.
- **Morfologia:** la morfologia aiuta a comprendere la natura delle parole, scomponendole in unità di significato più piccole dette "morfemi". Le parole che non possono essere scomposte in morfemi prendono il nome di "morfemi lessicali", mentre prefissi e suffissi, i quali si combinano con i morfemi lessicali, prendono invece il nome di "morfemi grammaticali". Ogni morfema consente di carpire delle informazioni sul significato della parola.
- **Analisi lessicale:** è lo studio che aiuta ad attribuire un significato alle parole quando queste vengono prese singolarmente.
- **Analisi sintattica:** attraverso questo tipo di analisi si cerca di analizzare le parole presenti all'interno di una frase, con l'obiettivo di comprendere la struttura grammaticale della frase stessa. I termini vengono analizzati al fine di identificarne la posizione e rivelare le eventuali dipendenze esistenti tra loro.
- **Analisi semantica:** è necessaria al fine di trovare il significato di una frase, contestualizzando le parole in essa contenute. Ogni parola può avere più di un significato, ma l'analisi semantica permette di identificarne il significato corretto, sulla base del contesto in cui la parola viene utilizzata.
- **Analisi pragmatica:** si effettua per riuscire ad estrapolare dalla frase dei significati che non sono letteralmente codificati, ad esempio quando il soggetto della frase è sottinteso.

La Natural Language Generation, invece, si trova all'opposto rispetto al Natural Language Understanding e si articola attraverso le tre fasi di identificazione degli obiettivi, pianificazione del loro raggiungimento e creazione di un testo che rappresenti quanto è stato pianificato.

Il NLP deriva dalla "Machine Translation" (MT), ossia un'area della linguistica computazionale che studia la traduzione di testi, da una lingua ad un'altra, mediante l'uso di programmi informatici. La MT riscosse molta attenzione a partire dalla fine del 1940, ma fu quasi del tutto abbandonata nel 1966, poiché molti la considerarono un vicolo cieco. Successivamente, negli anni 80', l'aumento della potenza computazionale dei processori e l'indebolimento delle teorie di Chomskyan, sulla linguistica, contribuirono alla nascita di algoritmi di machine learning per l'elaborazione del linguaggio naturale; ciò riuscì a dare nuova linfa alla MT e segnò la nascita del NLP. Grazie al machine learning, è stato introdotto l'uso di modelli statistici in luogo delle regole if-then scritte a mano, e questo ha costituito un vero punto di svolta, in quanto i software, utilizzando un approccio iterativo di apprendimento, riescono ad acquisire le regole del linguaggio in modo completamente autonomo, senza la necessità che queste siano codificate, una per una, a priori.

In generale, la capacità della mente umana di elaborare, comprendere e produrre il linguaggio è data da un insieme di meccanismi isolati. Allo stesso modo, il NLP si dirama in molti ambiti di ricerca diversi, detti "task", ciascuno dei quali assolve ad un compito specifico. Tale modularità permette sia l'utilizzo indipendente dei prodotti dei singoli task, sia la combinazione di alcuni di essi, al fine di giungere alla creazione di software che imitino nel miglior modo possibile il processo comunicativo umano. Tra i suddetti task i più rilevanti sono:

- Automatic Summarization: nell'era digitale, il sovraccarico delle informazioni costituisce un vero e proprio problema e la tendenza a produrre più informazioni di quante la mente umana sia possa comprenderne non sembra rallentare. Lo scopo dell'automatic summarization è quello di fornire il riepilogo di una grande quantità di dati, senza perdere però il loro significato.
- Co-Reference Resolution: costituisce una tecnica utilizzata al fine di individuare quali parole, all'interno di una frase, si riferiscono ad uno stesso oggetto.
- Machine Translation: si occupa della risoluzione di un problema antichissimo, ossia la traduzione da una lingua ad un'altra e, come già detto, ha costituito il punto di partenza per lo studio sulle tecniche di elaborazione del linguaggio naturale. La MT non ha come scopo principale la traduzione delle singole parole, ma punta a tradurre intere frasi, preservandone il significato intrinseco.
- Named Entity Recognition (NER): ha come obiettivo quello di individuare entità nominali (nomi di persona, organizzazioni, località, espressioni di tempo o quantità, etc.) e classificarle in categorie predefinite.
- Optical Character Recognition (OCR): è l'area dedicata al riconoscimento di caratteri in formato analogico e alla loro traduzione in un formato digitale, in modo tale da renderlo comprensibile alle macchine.
- Part Of Speech Tagging: anche indicato come "POS Tagging", consente di etichettare e classificare in modo efficiente le parole (nomi, aggettivi, verbi, etc.) all'interno di una frase. Le tecniche di POS Tagging variano a seconda della lingua utilizzata, ad esempio per la lingua Sanscrita viene utilizzata una tecnica chiamata "treebank", mentre per l'arabo ci si serve della "Support Vector Machine" o SVM.

Il principale problema che il NLP si trova a dover fronteggiare è costituito dall'ambiguità, dovuta al fatto che, in molte lingue, una stessa parola può avere più di un significato. Questo problema viene incontrato sia a livello sintattico, quando si tenta di interpretare intere frasi, che a livello lessicale e morfologico, quando si tenta di capire il senso di singole parole e sono state studiate diverse tecniche al fine di risolvere risolvere il problema; tra le possibili soluzioni proposte troviamo le seguenti: "Minimising Ambiguity", "Preserving Ambiguity", "Interactive Disambiguity" and "Weighting Ambiguity".

2. Software utilizzati

In questo capitolo verranno illustrati tutti gli strumenti software che sono stati utilizzati durante il lavoro svolto in Rai Pubblicità. Saranno dapprima descritti i programmi utili all'estrazione e alla presentazione dei dati, quali SQL Developer e Qlik Sense, dopodiché si procederà a descrivere la piattaforma di NLP necessaria allo sviluppo del chat bot e l'applicazione di messaggistica Telegram, usata per l'interazione con il bot stesso.

2.1 SQL Developer

SQL Developer è un ambiente di sviluppo, completamente gratuito ed altamente intuitivo, che permette di semplificare la creazione e la manutenzione dei database Oracle. Gli strumenti messi a disposizione dal programma possono essere d'aiuto sia agli sviluppatori, per eseguire query scritte in SQL e PL/SQL, sia ai database administrator (DBA) per la gestione ed il monitoring dell'intera base dati.

Le operazioni basilari supportate da SQL Developer comprendono la creazione e la gestione di connessioni verso i database di interesse, la visualizzazione degli oggetti propri di ogni connessione (tabelle, viste, etc.), l'esecuzione di query al fine di creare, modificare o cancellare gli oggetti, l'esportazione di sottoinsiemi di dati, la scrittura ed il debugging degli script. Oltre a queste, altre funzionalità possibili sono la migrazione su piattaforma Oracle dei dati salvati su altre piattaforme, lo sviluppo di applicazioni REST per la presentazione delle informazioni e la creazione di data model, grazie al "SQL Developer Data Modeler" (SDDM).

2.2 Qlik Sense

Qlik è un'azienda di software statunitense, fondata nel 1993 a Lund, in Svezia, che viene identificata come l'azienda leader in ambito business intelligence e data analytics. Tra i principali prodotti ideati da Qlik troviamo Qlik Sense, un'applicazione self-service, disponibile sia in versione desktop che in versione server, che consente la costruzione di un modello dati, a partire da fonti eterogenee, e la creazione di dashboard interattive per la visualizzazione delle informazioni.

La maggior parte degli strumenti moderni di BI utilizza SQL, adottando un approccio query-based, e questo costringe gli utenti ad una navigazione lineare, limitata a ben precisi sottoinsiemi dei dati disponibili, definiti sulla base di interrogazioni ed aggregazioni ipotizzate a priori. In questo scenario, i dati "imprevisti" vengono persi e qualora un utente dovesse aver bisogno di analizzare tali dati, si dovrebbero costruire nuove query ad hoc. Ogni nuova domanda necessita, quindi, di un periodo di attesa ed è per questo che tale modo di lavorare viene definito "ask, wait, answer cycle", ossia "ciclo domanda, aspetta, rispondi".

Qlik Sense, contrariamente a quanto visto fino ad ora, è dotato di un "motore associativo", il Qlik Associative Engine, e di un "motore di indicizzazione", il Qlik Indexing Engine (QIX), progettati al fine di consentire l'esplorazione e l'analisi interattiva dei dati, senza limitazioni dovute a query predefinite o a pre-aggregazioni. I miglioramenti che Qlik Sense propone rispetto ai sistemi convenzionali sono molteplici, ma ce ne sono tre su cui è necessario soffermarsi:

- Viene consentita la combinazione di un gran numero di sorgenti dati eterogenee tra loro, quali database relazionali, datawarehouse, file locali o altri. Internamente, Qlik Sense realizza una “full outer join” tra le sorgenti e ciò vuol dire che gli utenti hanno accesso a tutti i dati presenti all’interno delle fonti, anziché ad un sottoinsieme di essi definito dalle query, cosa che avveniva con i tradizionali strumenti di BI. In aggiunta a ciò, i dati vengono caricati anche se incompleti o non corretti, in quanto saranno rielaborati successivamente, per la costruzione del modello dati all’interno di Qlik.
- Il caricamento viene fatto “in-memory”, sfruttando un particolare algoritmo di compressione accelerata, che memorizza i dati estratti in due tipi di tabelle: “tabella dei dati” e “tabella dei simboli”. La tabella dei dati serve a rappresentare l’insieme delle informazioni estratte, ma anziché contenere i dati veri e propri, contiene soltanto dei puntatori. I puntatori servono a ricercare il valore corrispondente ad ogni dato, contenuto nella tabella dei simboli che, in pratica, funge da tabella di ricerca, creando al proprio interno una corrispondenza tra puntatori e dati. Non esiste un’unica tabella dei simboli, ma ne viene creata una per ogni tipo di dato da rappresentare. Il vantaggio del caricamento in-memory è dato dal fatto che le tabelle dei dati, contenendo dei “bit-stuffed pointer”, risultano avere una dimensione ridotta e ciò diminuisce il tempo necessario ad effettuare interrogazioni e aggregazioni, anche in presenza di grandi quantità di informazioni.
- Grazie al Qlik Associative Engine e al Qlik Indexing Engine, gli utenti possono effettuare delle selezioni sugli elementi grafici contenuti nelle dashboard e ad ogni loro click corrisponde un ricalcolo istantaneo del modello dati sottostante. Per ogni interazione, dunque, c’è un cambio di contesto che avviene “alla velocità del pensiero”, non rallentando gli utenti nella loro navigazione. Da un punto di vista grafico, inoltre, vengono sempre messe in evidenza non solo le associazioni esistenti tra i dati (evidenziate con il colore verde), ma anche quelle non consentite (evidenziate con il colore grigio). Tale comportamento viene detto “potenza del grigio” e permette la scoperta di nuove aree di opportunità o rischio.

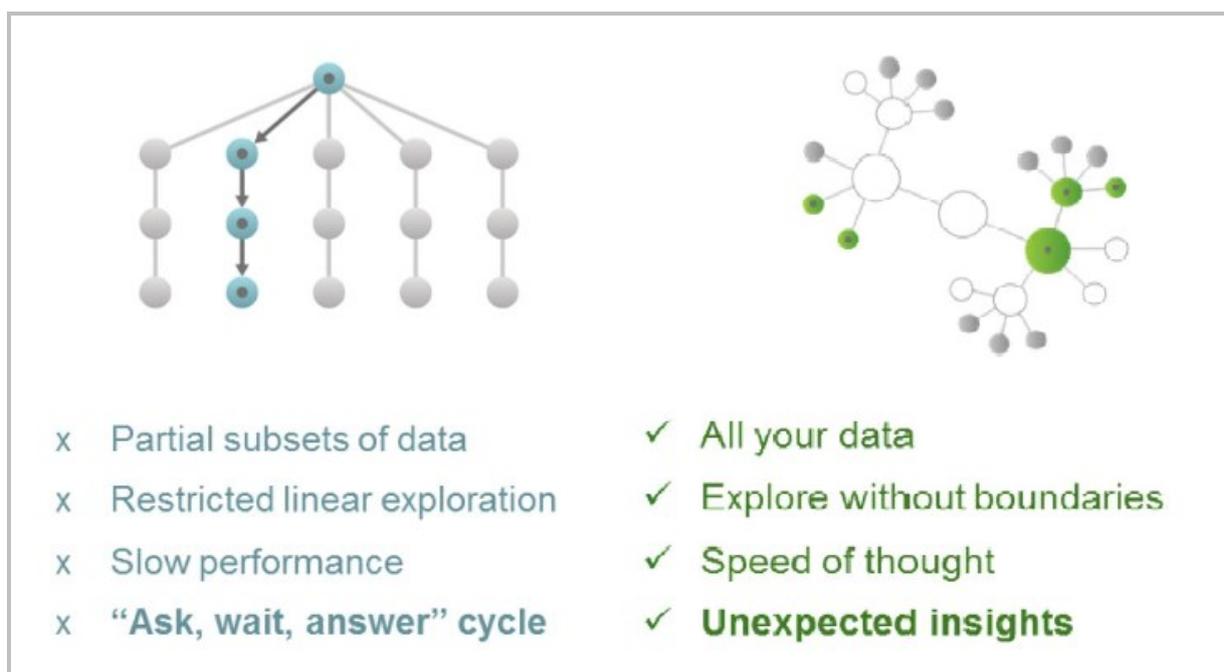


Figura 4 - Modello strumenti di BI tradizionali vs Modello associativo Qlik

Da un punto di vista pratico, Qlik Sense, sia nella versione desktop che in quella server, mette a disposizione degli strumenti per la creazione di applicazioni. Ciascuna applicazione ha precise finalità aziendali ed un proprio modello dati a cui attingere. In fase di presentazione, poi, ogni applicazione è composta da un certo numero di fogli, ossia dashboard interattive che sfruttano oggetti quali tabelle, grafici e kpi per rendere immediatamente fruibili i valori di interesse. Nella versione server, tutte le applicazioni sono inserite all'interno di uno spazio di lavoro condiviso detto "hub", in modo tale che più utenti possano lavorare sulla stessa app in parallelo. Gli accessi alle singole applicazioni sono regolati in base a permessi definiti dall'amministratore, mediante la Qlik Management Console (QMC), ossia una vera e propria area di amministrazione in cui è possibile configurare e monitorare le applicazioni sviluppate. In ultima analisi, le applicazioni Qlik Sense risultano essere un mezzo efficiente per l'esplorazione delle informazioni, in quanto lasciano gli utenti liberi di dare sfogo alle proprie idee e di seguire le proprie intuizioni, dovunque ed in qualsiasi momento, anche tramite smartphone o tablet.

2.3 Google Dialogflow

Dialogflow, conosciuto anche come Api.Ai, è un tool di proprietà di Google, pensato per lo sviluppo di strumenti finalizzati all'interazione uomo-macchina, basato su NLP e sostenuto da algoritmi di machine learning. Inizialmente, Api.Ai costituiva il motore alla base di "Assistant", un assistente virtuale multiplatforma, ideato dall'azienda Speaktoit, che eseguiva operazioni ed era in grado di rispondere alle domande degli utenti utilizzando il linguaggio naturale. Successivamente, a partire dal 2014, fu rilasciato anche agli sviluppatori e offriva supporto al riconoscimento vocale, al Natural Language Understanding e al text-to-speech. Nel 2016, Google rilevò la compagnia Speaktoit e, nel 2017, rinominò Api.Ai in "Dialogflow". Oggi i prodotti sviluppati attraverso Dialogflow funzionano sulla maggior parte dei dispositivi (telefoni, dispositivi wearable, macchine, speaker, etc.) e supportano oltre 14 lingue, inoltre, le operazioni di NLP vengono eseguite sempre sull'infrastruttura di Google e questo rende le applicazioni scalabili e utilizzabili anche da milioni di utenti. Di seguito saranno illustrate le principali componenti che contribuiscono al funzionamento di Dialogflow e, di conseguenza, allo sviluppo di interfacce testuali e vocali, basate sull'intelligenza artificiale.

- Agents: non sono altro che i moduli dedicati al Natural Language Understanding che, incorporati all'interno delle applicazioni, permettono di trasformare le richieste degli utenti in dati utilizzabili. Tale trasformazione si verifica ogni volta che una richiesta formulata corrisponde ad uno degli intent definiti all'interno dell'agent.
- Intents: come già detto, ogni volta che un utente effettua una richiesta, questa viene presa in carico dall'agent, il quale prova a trovare una corrispondenza con uno degli intent contenuti al suo interno. L'intent, dunque, costituisce una componente vitale per il funzionamento degli assistenti virtuali, in quanto crea un mapping tra le domande degli utenti e le "actions" che devono essere attuate conseguentemente. Per ogni intent, lo sviluppatore deve creare un dataset, contenente delle "training phrases", ossia delle frasi di esempio; più sarà grande il data set fornito, migliore sarà la comprensione di ciò che l'utente scriverà. Oltre a fornire la possibilità di creare nuovi intent, Dialogflow mette a disposizione intent predefiniti, al fine di semplificare la produzione di assistenti virtuali.
- Entities: possono essere assimilate a delle variabili e servono ad estrapolare le parole chiave contenute nelle richieste degli utenti. Per ogni agent viene definito un preciso insieme di entities sulla base dei parametri che ci si aspetta di ricevere ed ogni entity può

ricadere in uno dei seguenti gruppi: “system entities”, “developer entities” e “user entities”. Le system entities sono definite a priori da Dialogflow e servono ad indicare tipologie di dato quali, ad esempio, date e colori; developer entities e user entities, invece, vengono definite dallo sviluppatore ma, mentre le prime sono variabili generiche, le seconde fanno riferimento ad un utente specifica sessione utente.

- Actions: una volta trovata la corrispondenza tra la richiesta di un utente ed uno dei possibili intent, bisogna compiere lo step successivo, ossia indicare la action da eseguire, utilizzando, eventualmente, i parametri estratti dalla domanda.
- Fullfillments: sono dei blocchi di codice che è possibile eseguire in corrispondenza di un determinato intent, al fine di realizzare particolari operazioni. I fullfillments non sono altro che l’implementazione di un web service che lavora sull’infrastruttura cloud di Google.
- Context: è uno degli elementi che fanno parte dell’intent e viene utilizzato per far sì che sia tenuta traccia delle precedenti richieste dell’utente, in modo tale che le richieste successive possano essere opportunamente contestualizzate. Il contesto è importante soprattutto per quelle frasi che risultano vaghe o che possono avere significati diversi; ad esempio, la frase “vorrei ascoltare un loro brano”, presa così com’è, sarebbe difficile da interpretare, ma, se fosse pronunciata mentre si discute di una band musicale, il suo significato risulterebbe molto più chiaro.

Una volta identificati gli attori che prendono parte al processo di elaborazione del linguaggio naturale in Dialogflow, è possibile passare all’analisi del percorso che porta dalla ricezione della richiesta utente, fino all’esecuzione di una action. Per prima cosa l’utente, attraverso una delle piattaforme supportate da Api.Ai, inoltra la richiesta all’agent, il quale la confronta con i possibili intent, al fine di trovare una corrispondenza. Dopo aver trovato l’intent corretto, vengono estratte e poi inviate al back-end tutte le entities contenute nella richiesta. Il back-end effettua delle elaborazioni che rispondono a logiche esterne, dopodichè manda la risposta calcolata a Dialogflow che, a sua volta, la recapita all’utente.

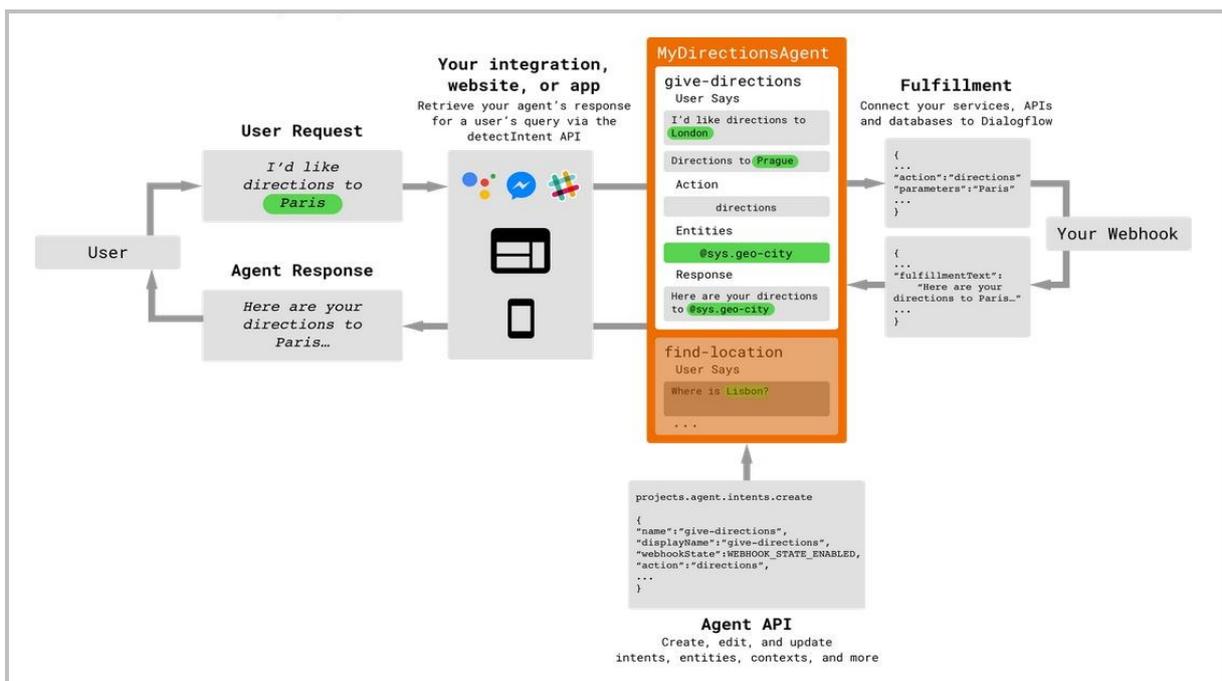


Figura 5 - Schema riassuntivo del funzionamento di Dialogflow

Il match tra richiesta utente ed intent viene individuato grazie all'utilizzo di algoritmi di machine learning. A partire dalle training phrases, infatti, Dialogflow sviluppa un modello (algoritmo) utile a capire, di volta in volta, quale sia l'intent corretto da sollevare; tale modello è diverso per ogni agent e viene aggiornato ogni volta che un utente fornisce un nuovo input. Le richieste effettuate, assieme alle training phrases già presenti all'interno del dataset, costituiscono la base per addestrare il sistema, andando a modificare in modo dinamico il modello. In fase di configurazione dell'agent, viene data la possibilità di scegliere il tipo di matching da utilizzare, che può essere "Hybrid" o "ML only". Nel primo caso è prevista la combinazione di tecniche di machine learning e regole e tale soluzione è consigliata per gli agent che non dispongono di un dataset ampio; l'approccio "ML only", al contrario, dovrebbe essere utilizzato solo se, per ogni intent, è stato fornito un gran numero di training phrases. Uno dei problemi che è possibile riscontrare nell'interpretare gli input è costituito dai cosiddetti "falsi positivi" ed è per questo motivo che Dialogflow consente anche di specificare, per ogni agent, un valore di soglia detto "classification threshold". Ogni coppia richiesta utente – intent totalizza un punteggio basato sul proprio livello di confidenza e se tale punteggio risulta inferiore al valore di soglia specificato, allora l'intent può essere sollevato.

Un'ulteriore feature di Dialogflow è la possibilità di monitorare gli agents attraverso la pagina "Analytics", ossia una sezione che consente di studiare il comportamento degli agents e di comprendere come migliorarli. La pagina di analisi presenta tre elementi: un grafico delle sessioni per rappresentare l'interazione giornaliera tra l'agent e l'utenza, una tabella delle query, contenente informazioni sugli intent (in quante sessioni l'intent è stato richiamato, quante volte quello è stato l'intent di chiusura della conversazione, etc.), e un "session flow" per mettere in evidenza l'ordine con cui gli intent vengono eseguiti quando gli utenti interagiscono con il sistema.

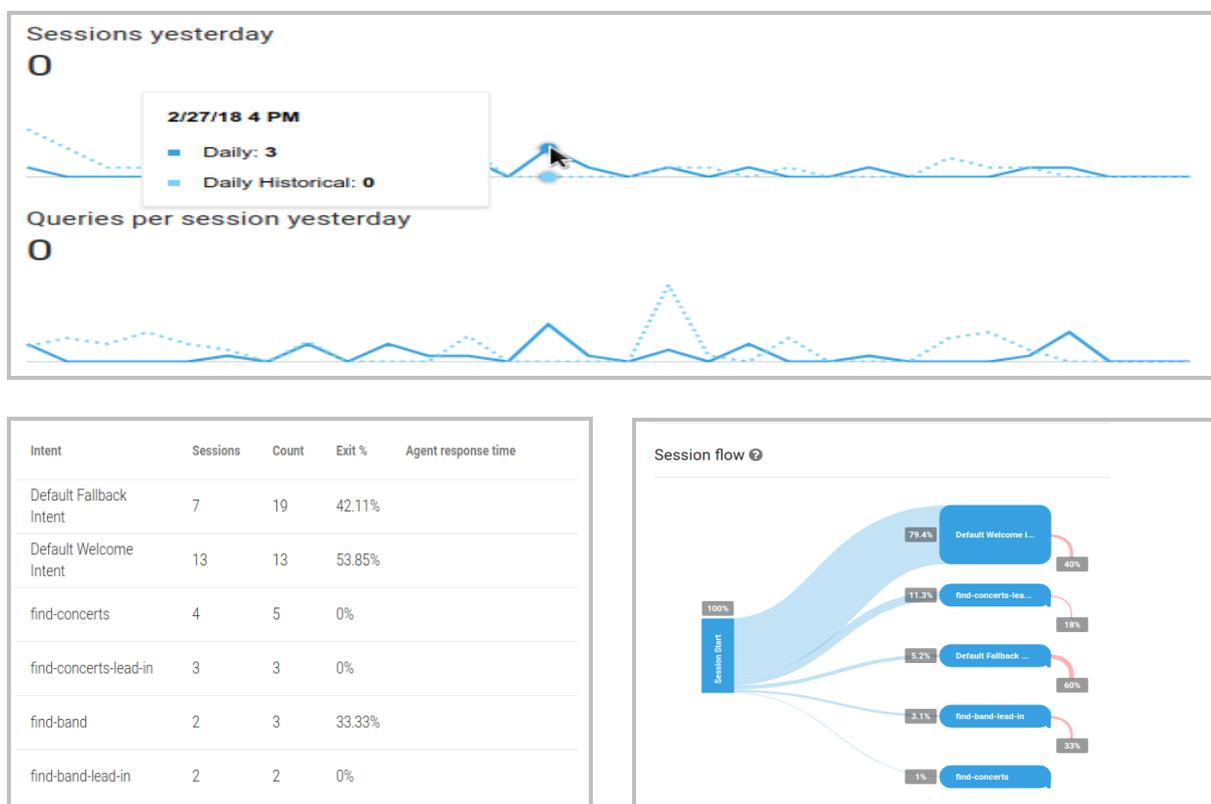


Figura 6 - Dialogflow Analytics

2.4 Telegram

Telegram, il cui slogan è “A new era of messaging”, è una piattaforma di messaggistica istantanea basata su cloud, ideata dalla società Telegram LLC. Attraverso l’app client, disponibile gratuitamente per tutte le piattaforme, è possibile stabilire conversazioni testuali e chiamate point-to-point, su canali cifrati; è inoltre consentito lo scambio di contenuti multimediali, messaggi vocali e file di qualsiasi tipo, fino ad una dimensione massima pari a 1,5 GB. Telegram è stata fondata da Nikolai e Pavel Durov nel 2013 e nel 2016 aveva già raggiunto quota 100.000.000 di utenti attivi mensilmente. Lo scambio di messaggi si basa su un protocollo proprietario, ideato proprio da Nikolai, chiamato MTProto, che prevede il salvataggio dei messaggi sul cloud, così da garantire la sincronizzazione istantanea e contemporanea di tutti i dispositivi cui l’utente è connesso. In realtà, oltre alla comunicazione standard, in cui, come già detto, i messaggi vengono salvati sul cloud ed è presente una cifratura client-server, esiste un tipo di comunicazione “segreto”, il quale prevede che i messaggi non vengano memorizzati sul cloud e che la cifratura sia end-to-end.

A partire dal 2015, Telegram ha aggiunto alle proprie funzionalità, la possibilità di creare “canali” e “bot”. I canali sono delle chat in cui l’amministratore è il solo a poter inviare i messaggi, che vengono ricevuti da tutti gli altri membri del canale; i bot, invece, sono degli account, gestiti in maniera automatizzata, che permettono l’esecuzione di applicazioni di terze parti all’interno di Telegram. Per lo sviluppo dei bot, sono state rese disponibili delle API, chiamate “Bot API”, che permettessero agli sviluppatori di integrare i loro back-end, eseguiti su server esterni, con il front-end di Telegram.

Sono tre gli elementi vitali per il funzionamento di un bot e cioè il client, ossia colui il quale contatta il bot attraverso l’app, il bot, ossia un software in esecuzione su un server esterno, ed il server di Telegram, che funge da intermediario tra i due. La prima parte della comunicazione si svolge quindi tra il client ed il server di Telegram, sfruttando il protocollo MTProto, dopodiché, il server Telegram, utilizzando il protocollo HTTPS, indirizza i messaggi ricevuti dal client verso il server esterno.

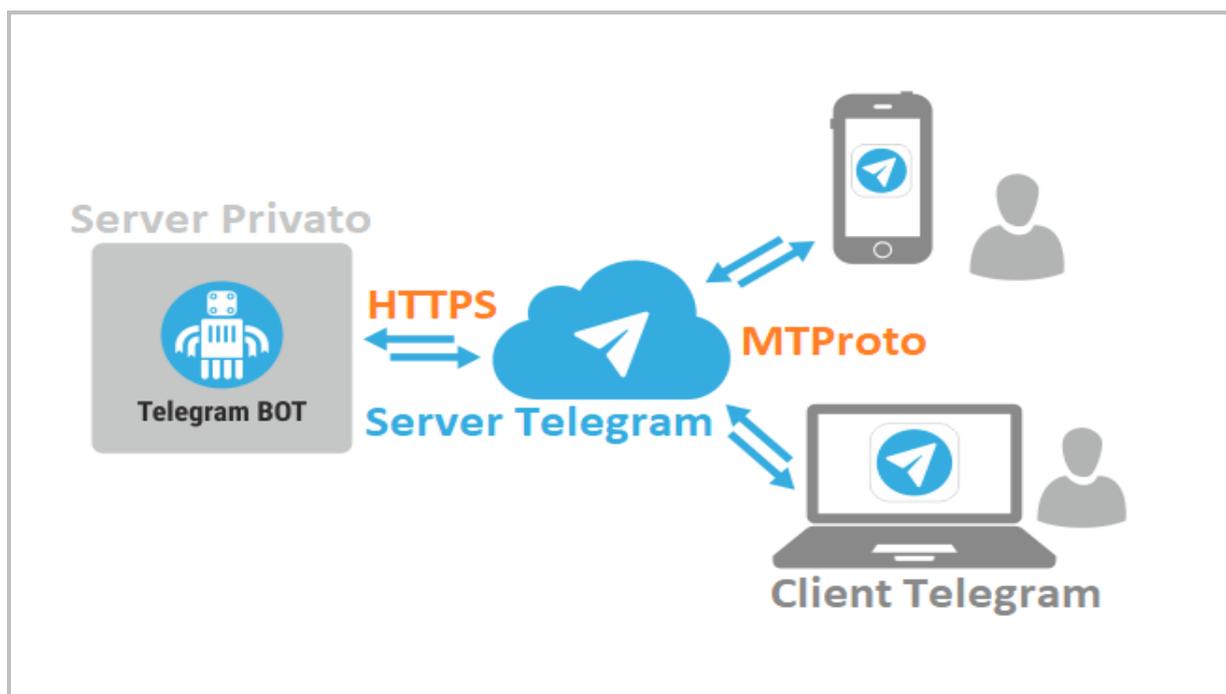


Figura 7 - Architettura di un generico Chat Bot Telegram

Per far sì che il bot riceva i messaggi da parte del server di Telegram, è possibile scegliere tra due strade possibili, ossia “webhooks” e “long polling”. Nel primo caso, l’indirizzo ip del server che ospita il bot è conosciuto dal server di Telegram e, ogniqualvolta quest’ultimo riceve un messaggio diretto al bot, effettua una richiesta HTTPS POST all’indirizzo ip del bot, il cui contenuto è il messaggio serializzato in JSON. Secondo l’approccio “long polling”, invece, il server che ospita il bot si comporta come se fosse un client e contatta periodicamente il server di Telegram per sapere se ci sono nuovi messaggi indirizzati a lui. Il “long polling” è chiaramente meno efficiente, ma consente la realizzazione di un bot anche nel caso in cui non si disponga di un server con ip pubblico.

3. Sviluppo delle app “AQC” e “Flow”

Oggetto di questo capitolo sarà il lavoro svolto in azienda al fine di realizzare, su piattaforma Qlik Sense, due applicazioni utili al monitoraggio degli investimenti della clientela. Innanzitutto, verrà spiegato con quale obiettivo le applicazioni siano state sviluppate, evidenziando anche quelle che erano le specifiche inizialmente fornite, dopodiché, si procederà ad una descrizione degli strumenti hardware e software aziendali e degli step generalmente previsti per la produzione di app Qlik Sense. Una volta terminata questa trattazione generica, si entrerà nel dettaglio delle applicazioni “AQC” e “Flow”, illustrando la struttura dei modelli dati che stanno alla base di queste due applicazioni e, soprattutto, focalizzando l’attenzione sulle tecniche utilizzate per creare le dashboard nel front-end.

3.1 Specifiche e pianificazione del lavoro

Quando ci si trova in ambito aziendale, è buona norma, prima di iniziare un lavoro, definire come tale lavoro sarà articolato, in modo da avere un’idea ben chiara dei tempi necessari per portarlo a termine e degli step da compiere affinché siano rispettate eventuali scadenze. Nell’ambito della BI, inoltre, risulta necessario che chi commissiona il lavoro chiarisca l’obiettivo finale che si vuole raggiungere e fornisca un elenco delle informazioni che dovranno essere rappresentate.

Relativamente al lavoro oggetto di questa tesi, l’utenza finale aveva commissionato al team di BI la produzione di un’applicazione che potesse fornire supporto in fase di trattativa con i clienti idonei alla stipula di un “accordo quadro”, ossia “un accordo concluso tra una o più stazioni appaltanti e uno o più operatori economici, il cui scopo è quello di stabilire le clausole relative agli appalti da aggiudicare durante un dato periodo in particolare per quanto riguarda i prezzi e, se del caso, le quantità previste”³. In aggiunta a ciò, l’utenza finale aveva fornito i file Excel “MASTER scheda per accordi quadro.xlsx” e “AQC – Dati per layout.xlsx”.

Nel primo file vengono elencate, in modo dettagliato, tutte le informazioni da esporre nelle dashboard, suddivise in base all’ambito di applicazione. Tali informazioni possono essere “complete”, se esistono già all’interno dei database aziendali, o “calcolate”, se non esistono ancora e vanno ricavate mediante apposite formule (specificate anch’esse nel documento). In generale, i dati di interesse riportati nel foglio elettronico, fissati un singolo cliente o una holding, comprendono:

- Informazioni sui venditori e sul settore merceologico di riferimento
- Informazioni sugli investimenti effettuati su Rai Pubblicità (fatturato totale, percentuale di sconto, sanatorie, etc.)
- Informazioni sugli investimenti effettuati sui principali competitors (quota Rai, quota Mediaset, quota media, etc.)
- Informazioni sugli investimenti relativi ai singoli prodotti, con dettaglio settimanale
- Informazioni sugli investimenti effettuati su TV e radio (GRP, CPG, etc.)

³ Articolo 3, comma 13, decreto legislativo n.50/2016, Codice dei contratti pubblici

– Informazioni sugli investimenti effettuati sul web (CPM, viewability, etc.)

Il secondo file, invece, propone una possibile strutturazione delle dashboard e riporta alcuni oggetti grafici da utilizzare come traccia, durante la fase di sviluppo. Sommarariamente, tale prototipo suggerisce di dedicare la prima schermata dell'app alla scelta del cliente (o della holding) e di realizzare, poi, una schermata che possa fungere da hub per la navigazione dell'applicazione. Gli altri fogli di esempio, esposti in questo file, fanno riferimento alle aree già identificate all'interno del documento "MASTER scheda per accordi quadro.xlsx", ossia "investimenti Rai Pubblicità", "investimenti competitors", "focus TV e radio", "focus web" e "andamento settimanale degli investimenti".

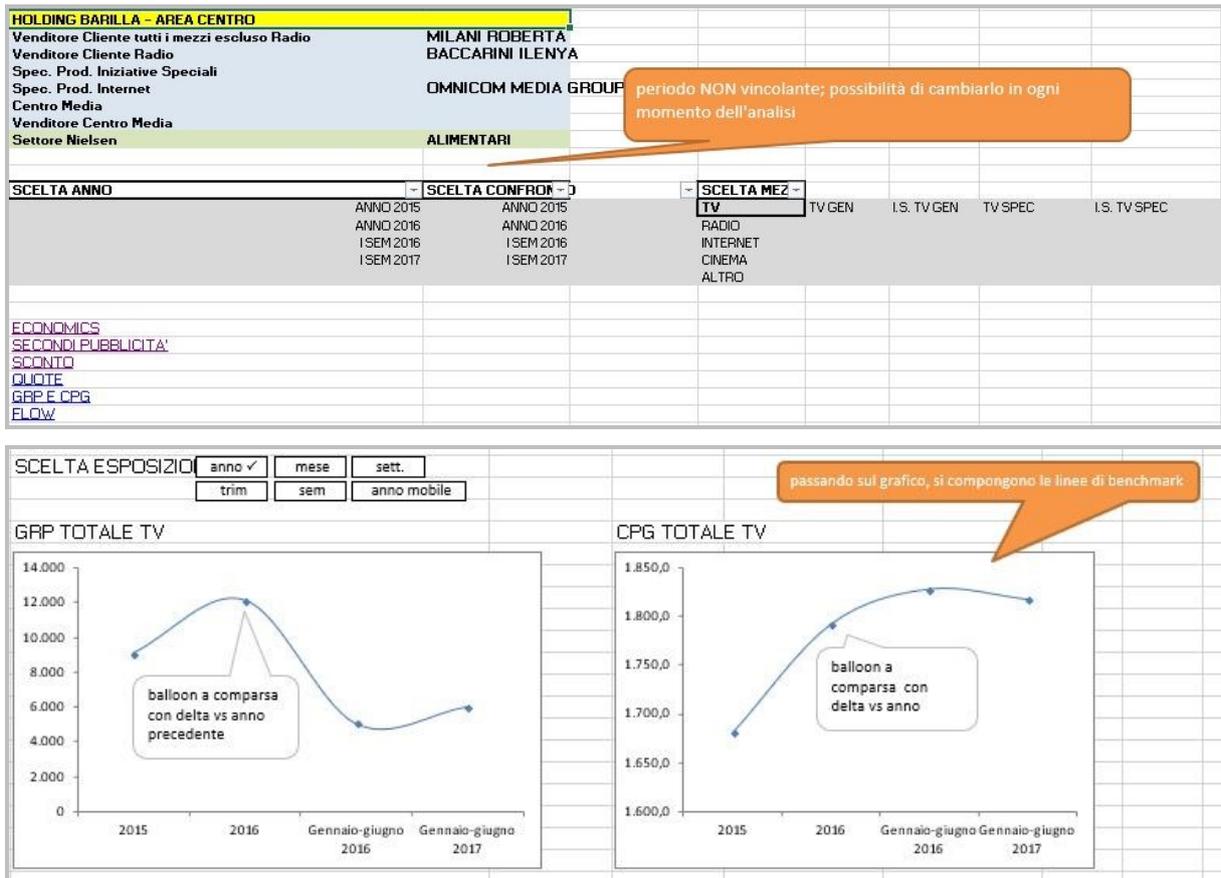


Figura 8 - Immagini estratte dal file "AQC - Dati per layout.xlsx"

Una volta prese in esame tutte le specifiche, è stato possibile delineare i passi successivi da compiere e si è deciso di procedere nel seguente modo: identificare le fonti contenenti le informazioni di interesse e creare strutture dati appositamente ideate per essere caricate in Qlik Sense, configurare l'ETL per automatizzare il caricamento dei dati in Qlik, costruire il modello dati a supporto dell'app e, infine, creare le dashboard interattive da consegnare all'utenza finale.

Nonostante fosse stata richiesta un'unica applicazione, si è deciso di utilizzare un'app secondaria per la rappresentazione del flow, ossia di quegli investimenti, inerenti ai singoli prodotti, dettagliati per settimana, in quanto, proprio a causa del dettaglio settimanale, tali informazioni devono poter scendere ad un livello di aggregazione diverso rispetto a tutte le altre.

3.2 Sistema aziendale di BI

Il sistema di BI aziendale ruota attorno alla piattaforma Qlik Sense, che viene utilizzata non solo per la produzione di app, finalizzate alla rappresentazione delle informazioni, ma anche per la realizzazione del framework ETL, utile ad automatizzare la fase di caricamento dei dati in Qlik, e per definire i criteri di accesso alle informazioni.

Dal punto di vista dell'hardware, Qlik Sense (v. 11.14.4) è installato su due server fisici distinti, aventi le stesse caratteristiche (4 x CPU Intel Xeon e7-8850 v2 @ 2.30GHz, RAM da 96 GB, sistema operativo Windows Server 2012, hard disk da 580 GB), in modo da dare vita a due ambienti di lavoro separati, chiamati "collaudo" e "produzione". L'ambiente di collaudo serve ad ospitare i progetti durante la loro fase di sviluppo, mentre l'ambiente di produzione viene utilizzato per contenere le app già revisionate e pronte all'uso. Oltre al server fisico, l'ambiente di produzione si serve anche di un NAS da 250 GB, utilizzato principalmente per la memorizzazione di fonti dati estemporanee, come file excel o documenti di testo.

Le informazioni caricate in Qlik Sense provengono per lo più dai database aziendali di BI, gestiti mediante un sistema Oracle RDBMS (Relational DataBase Management System). Il software per la gestione della base dati (Oracle Database v. 11g) è installato su quattro server fisici, così da avere quattro basi dati distinte, ciascuna delle quali coincide con uno dei seguenti ambienti:

- Sviluppo: questo ambiente è utilizzato per sperimentare nuove soluzioni, creando, eliminando o modificando tabelle e viste, senza che eventuali errori si ripercuotano sul prodotto finale. Rispetto agli altri ambienti, sviluppo ospita solo una parte delle informazioni utili, in modo che tutte le elaborazioni effettuate risultino veloci.
- Collaudo: una volta che la fase di sviluppo è stata completata, gli oggetti realizzati vengono spostati in collaudo e ne viene permesso l'uso da parte degli utilizzatori finali, in modo che questi possano interagirvi ed accorgersi di eventuali problemi da risolvere, prima del rilascio in produzione. Il database su cui si erge l'ambiente di collaudo non contiene più solo una parte delle informazioni aziendali, ma l'intero patrimonio dati.
- Produzione: è l'ambiente sotto il quale vengono spostati tutti gli oggetti che hanno superato la fase di collaudo. Per quanto riguarda i dati, non ci sono differenze tra collaudo e produzione.
- Test: per l'ambiente di test si può considerare quanto già detto per l'ambiente di sviluppo, con la differenza che in test sono presenti più informazioni.

Ognuno degli ambienti appena visti suddivide i dati nelle seguenti tre aree, realizzando di fatto tre database logici su ogni server fisico:

- DB Commerciale: comprende i dati "anagrafici" relativi agli ambiti TV, radio, cinema, web, fatturazione, venditori, etc.
- DWH o Datawarehouse: comprende, oltre ad una copia delle informazioni già presenti all'interno del DB Commerciale, anche i dati sulle vendite interne, raccolti da Rai Pubblicità, e quelli sulle vendite dei principali competitor, raccolti dalla società Nielsen.
- SAP: ospita i dati relativi al mondo della contabilità.

I due ambienti Qlik di collaudo e produzione caricano i dati, rispettivamente, dagli ambienti Oracle di collaudo e produzione, collegandosi ad essi mediante una connessione TNS (Transparent Network Substrate). Per accedere ai database Oracle è sempre necessario autenticarsi attraverso username e password, in quanto ogni utente può operare solo su un

insieme ridotto di oggetti, diverso da utente ad utente e definito a priori. Nel caso specifico di Qlik, per il collegamento ai database sono state definite due utenze, chiamate “QlikUSR” e “QlikADM”; la prima consente l’accesso a tutte le viste necessarie alla creazione delle app in Qlik Sense, mentre la seconda permette di accedere alle tabelle utili alla configurazione dell’ETL.

3.2.1 Il framework ETL aziendale

Il framework ETL costituisce l’anello di congiunzione tra i database Oracle e le applicazioni Qlik Sense e presenta come principale vantaggio quello di velocizzare il processo di caricamento dei dati all’interno delle applicazioni. Il prodotto finale dell’ETL, infatti, è un insieme di file QVD (Qlik View Data), che le app utilizzano come sorgente per le informazioni; tale formato è nativo di Qlik e permette un caricamento fino a cento volte più veloce, rispetto a quello effettuato direttamente sulle fonti. La struttura scelta per la realizzazione dell’ETL è quella “4 Tier QVD Based”, cioè una struttura che sfrutta, complessivamente, quattro livelli ETL in cascata; due per il caricamento delle informazioni comuni a tutte le app e altri due per la creazione dei QVD dedicati alle singole applicazioni.

Il primo livello viene detto “ETL 01 centralizzato” e non fa altro che estrarre i dati utili dalle sorgenti (DB Commerciale, DWH, SAP) e caricarli all’interno di file QVD, senza alterarli in alcun modo. I QVD così ottenuti vengono memorizzati all’interno di un’area di “staging”, prima di subire le elaborazioni proprie dei livelli successivi.

Il secondo livello viene detto, invece, “ETL 02 centralizzato” e ha l’onere di prelevare i dati dai QVD presenti all’interno dell’area di staging, per applicarvi le trasformazioni che risultano utili a tutte le applicazioni. Allo stesso modo dell’ETL 01 centralizzato, i dati ottenuti a questo livello vengono memorizzati in dei file QVD, dopodiché sono conservati nel cosiddetto “common data layer”.

Il terzo livello prende il nome di “ETL 01 di app” e viene utilizzato al fine di estrarre, dai QVD e da fonti estemporanee, le informazioni utili alle singole applicazioni, per poi caricarle in nuovi file QVD, memorizzati all’interno di directory dedicate ai singoli progetti.

L’ultimo livello è chiamato, intuitivamente, “ETL 02 di app” ed è necessario per effettuare ulteriori trasformazioni sui dati, qualora queste siano richieste dall’applicazione cui l’ETL fa riferimento. I QVD risultanti a questo livello vengono memorizzati anch’essi in directory dedicate ai singoli progetti e costituiscono la base cui attingono le applicazioni Qlik vere e proprie. Risulta opportuno sottolineare che le operazioni attuate sui dati a questo livello sottendono già il modello dati che si andrà a formare, in modo automatico, una volta caricate le informazioni nell’app finale.

Come già affermato all’inizio di questo paragrafo, il framework ETL è realizzato in Qlik e questo vuol dire che i quattro livelli appena descritti sono anch’essi applicazioni Qlik Sense che, basate su script predefiniti, realizzano le operazioni di estrazione, trasformazione e caricamento dei dati. Le app relative all’ETL vengono ricaricate, in modo automatico, una volta alla settimana, grazie al Qlik Sense Scheduler Service (QSS) ed il caricamento segue un ordine ben preciso: prima viene eseguito il codice corrispondente ai livelli ETL 01 ed ETL 02 centralizzati, dopodiché, per ogni progetto, vengono eseguiti gli script di caricamento propri di ETL 01 ed ETL 02 di app e, infine, viene caricata l’app vera e propria (Front-End). L’operazione di caricamento fa sì che i dati esposti nel Front-End, per mezzo delle dashboard, risultino allineati rispetto a quelli presenti all’interno dei database aziendali e la forza dell’ETL sta proprio nel rendere questo processo di allineamento automatico, centralizzato e scalabile. In caso di necessità particolari, inoltre, è sempre possibile effettuare il caricamento delle app, manualmente, senza dover attendere il giorno della settimana prestabilito.

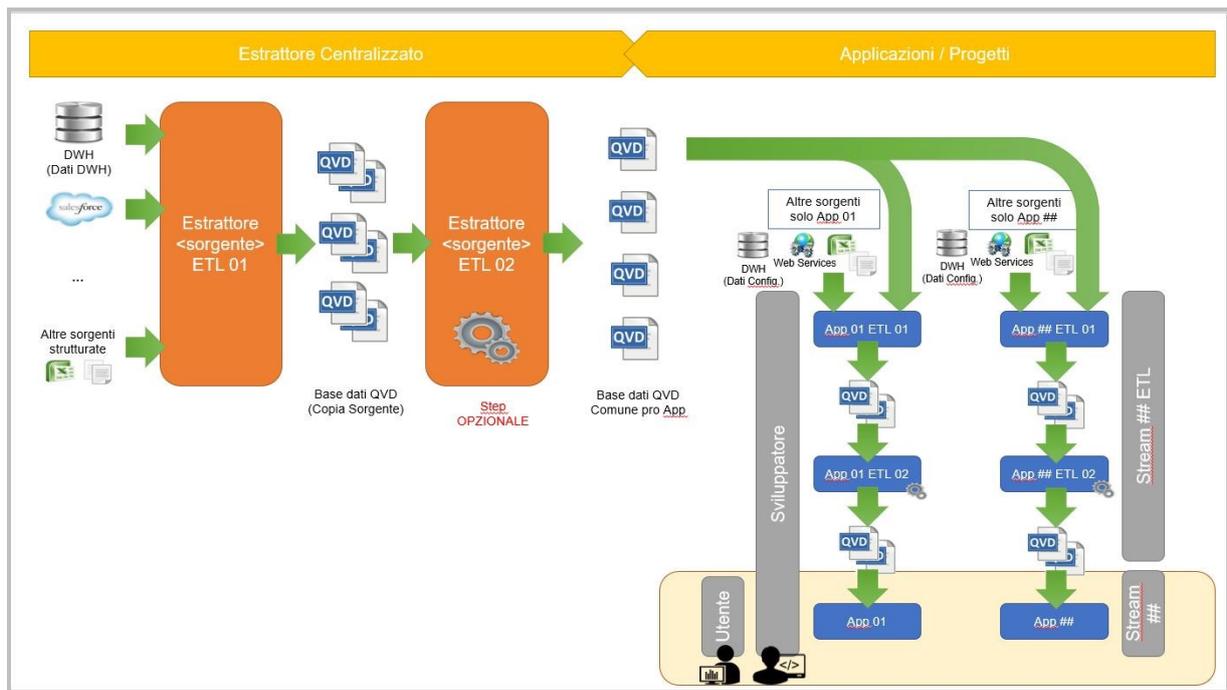


Figura 9 - Architettura ETL aziendale “4 Tier QVD Based”

Le app di ETL, per semplicità, utilizzano tutte il medesimo script per il caricamento dei dati, ma generano una parte del codice in modo dinamico, servendosi dei parametri presenti all'interno delle tabelle di configurazione. In questo modo, pur partendo da una base comune, riescono a differenziare non solo il comportamento dei diversi livelli, ma anche il modus operandi degli ETL relativi a progetti diversi. Dunque, popolare le tabelle di configurazione in modo opportuno costituisce il cuore del processo di creazione del framework ETL. Tali tabelle vengono create sotto il profilo QlikADM e di seguito ne viene fornita una descrizione dettagliata:

- ETL_GLOBAL_VARIABLES è la tabella utilizzata per ospitare un insieme di variabili utili a tutti gli ETL. Solitamente nella tabella in questione sono presenti i path in cui è possibile trovare i dati sorgente e i path in cui andare a memorizzare i QVD prodotti.
- ETL_01_CONNECTION è la tabella atta a contenere le stringhe di connessione ai database relazionali e alle fonti dati estemporanee.
- ETL_01_TABLETOEXTRACT è la tabella contenente l'elenco delle tabelle da caricare, evidenziando le eventuali condizioni di “where”, necessarie al filtraggio delle informazioni.
- ETL_01_QUERY è la tabella usata per memorizzare l'elenco di campi che, per ogni tabella indicata in ETL_01_TABLETOEXTRACT, devono essere caricati, specificando se tali campi vanno rinominati o se, invece, vanno calcolati a partire dalle informazioni esistenti.
- ETL_01_VARIABLES è la tabella utilizzata per ospitare le variabili non globali, utili al singolo ETL di primo livello (centralizzato o di app). Tali variabili vengono richiamate durante l'esecuzione dello script, dopodiché il loro valore torna quello iniziale.
- ETL02_INDEX è una tabella analoga a ETL_01_TABLETOEXTRACT e contiene l'elenco delle tabelle da caricare a partire dall'area di staging.

- ETL02_TABLES è una tabella analoga a ETL_01_QUERY e contiene l’elenco dei campi da estrarre da ognuna delle tabelle presenti in ETL02_INDEX, indicando per ognuno di questi se sia necessaria una rinomina o eventuali altre elaborazioni.
- QLIK_CATALOGO_INFORMAZIONI è, alla stregua di ETL_GLOBAL_VARIABLES, una tabella globale che contiene, per quei campi del database che necessitano di una rinomina, l’alias da utilizzare. La QLIK_CATALOGO_INFORMAZIONI, inoltre, contiene un flag che, per ogni campo, indica se tale campo debba risultare visibile nell’applicazione finale o meno.

Dal momento che le tabelle appena analizzate sono comuni a tutti i livelli di ETL, in ogni tabella è presente una colonna denominata “PROJECTID” che indica, per ciascun record, l’ETL per cui quella riga è di interesse. Il valore che è possibile trovare all’interno del campo “PROJECTID” è pari a 1, se il record è di interesse per uno dei due ETL centralizzati, compreso tra 101 e 999, se il record è di interesse per uno degli ETL di app presenti in produzione, uguale o superiore a 1000, se il record è di interesse per uno degli ETL di app presenti in collaudo.

3.2.2 Controllo degli accessi

Una volta attraversati tutti gli ETL che compongono il sistema di BI aziendale, le informazioni possono essere finalmente organizzate all’interno delle dashboard e diventano fruibili agli utenti finali. Nasce a questo punto l’esigenza di profilare gli utenti, affinché la visibilità delle informazioni sia limitata soltanto a coloro che ne hanno effettivamente diritto.

Qlik consente, mediante il pannello di amministrazione, di creare un elenco di utenti “autorizzati”, identificati mediante user id, ai quali può essere conferito il grado di amministratore o di utente semplice. Ogniqualvolta un utente prova ad accedere a una delle app Qlik Sense presenti in collaudo o in produzione, una componente presente sul server Qlik, chiamata “Qlik Sense Proxy” (QSP), provvede ad autenticarlo, utilizzando la tecnologia NTLM. Gli utenti Rai Pubblicità, per poter utilizzare i pc aziendali, devono loggarsi usando la propria utenza Windows e, con NTLM, questa stessa utenza viene utilizzata per autenticare gli utenti anche su Qlik Sense, in modo totalmente automatico.

Tramite il pannello di amministrazione è anche possibile definire dei vincoli di accesso che fanno riferimento ai singoli stream o alle singole app disponibili. Nello specifico, viene data la possibilità di creare le cosiddette “security rules”, ossia regole che indicano, per ogni oggetto (app, stream, etc.), quali tipologie di utenti siano autorizzati ad accedere e quali no. Le security rules sono basate su brevi script che contengono riferimenti a ben specifiche proprietà degli oggetti, definite a-priori, chiamate “custom properties”. Ogniqualvolta un utente tenta l’accesso ad una risorsa presente in Qlik, il sistema valuta tutte le security rules che sono state definite per quella risorsa e, solo in seguito a tale verifica, decide se concedere l’accesso all’utente.

Security rules e custom properties, però, non tornano utili al fine di capire, all’interno delle applicazioni, quali sono le informazioni cui ogni utente è autorizzato ad accedere, ma effettuano la gestione degli accessi ad un livello più alto. Per quanto riguarda la visibilità che gli utenti hanno a livello dei dati, invece, esistono due approcci possibili, che prendono, rispettivamente, i nomi “Segregazione Orizzontale” e “Segregazione Verticale”. Mentre la segregazione orizzontale comporta una riduzione dei record (righe) che ogni utente può visualizzare, quella verticale comporta una limitazione dal punto di vista dei campi (colonne), ma entrambe le soluzioni vengono realizzate grazie ad un apposito script, inserito all’interno delle app Qlik Sense, che viene comunemente indicato come “Section Access”. Il codice relativo alla section access non è altro che un template standard il quale, come già visto per gli

ETL di caricamento, viene particolarizzato, andando a recuperare, in modo dinamico, una serie di parametri contenuti nei database aziendali, in tabelle compilate sotto il profilo QlikADM. In generale, per quanto riguarda la segregazione orizzontale, viene utilizzato un approccio di tipo “whitelisting”, ossia viene indicato per ogni cluster di utenti (clienti, agenzie, Strutture vendita, Tipi prodotto pubblicitario) l’elenco dei campi visibili. Per la segregazione verticale, invece, il discorso è diverso, in quanto viene specificato, per ciascun utente, l’elenco delle informazioni da nascondere (Omit Group), andando a prediligere una soluzione di tipo “blacklisting”.

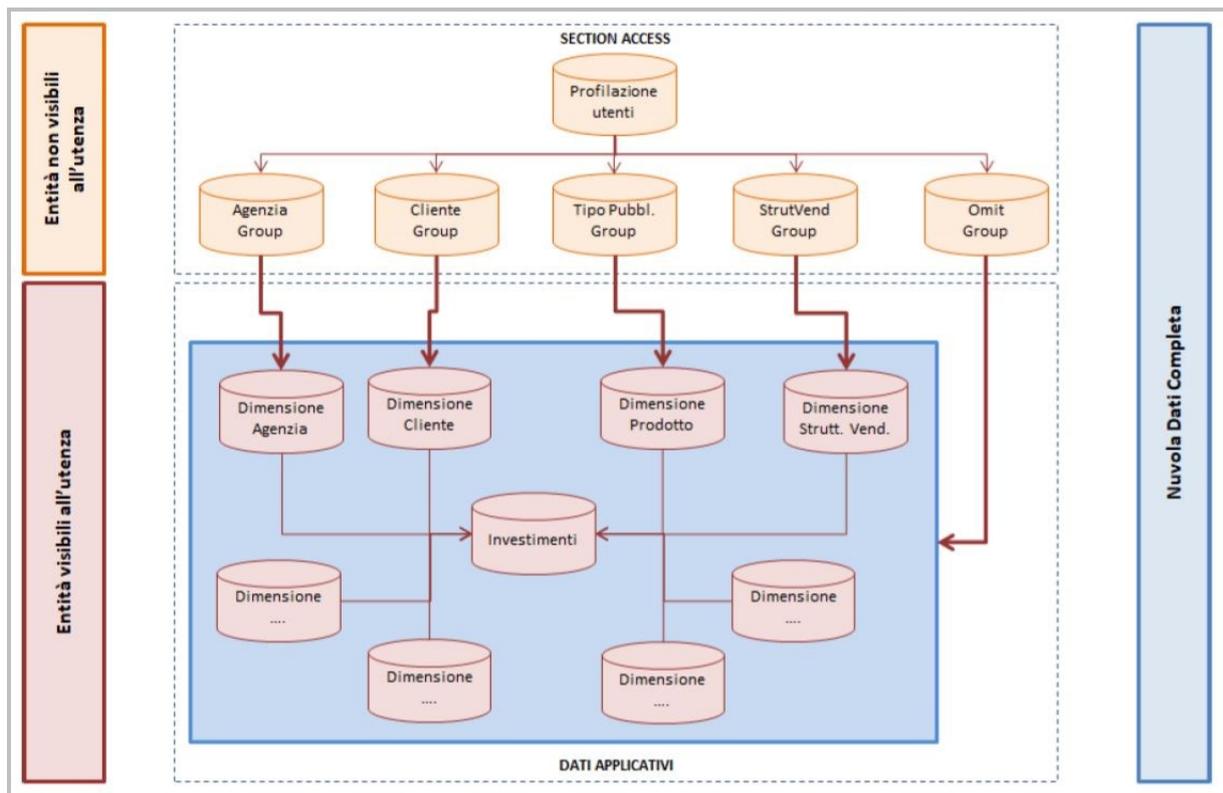


Figura 10 - Segregazione orizzontale e verticale dei dati nel sistema Rai Pubblicità

3.3 Reperimento delle informazioni utili

Una volta analizzate le specifiche richieste da parte dell’utenza che ha commissionato il lavoro e compreso il funzionamento del sistema attraverso il quale avviene la produzione degli applicativi Qlik Sense, si è dato inizio allo sviluppo delle app “AQC” e “Flow”, partendo da quello che è lo step iniziale, ossia la ricerca delle fonti dati utili all’interno dei database aziendali.

Durante la fase di ricerca, ci si è accorti che i dati necessari alle app erano contenuti, quasi completamente, all’interno delle tabelle compilate sotto l’istanza DWH che, come detto in precedenza, ospita sia le informazioni sugli investimenti dei clienti, sia le anagrafiche ad esse connesse. Però, se da una parte è stato possibile rinvenire i dati sugli investimenti e sulle anagrafiche, dall’altra è stata appurata la mancanza di alcune informazioni inerenti al mondo TV e radio (GRP e CPG) e al mondo web (Viewability e Posizioni pregiate) e ciò ha reso necessaria la richiesta di creazione di nuove tabelle che contenessero tali informazioni, sempre sotto l’istanza DWH.

A causa del tempo necessario alla produzione delle suddette tabelle, incompatibile con la data prevista per la consegna delle app, si è deciso di adottare una soluzione “momentanea”, predisponendo delle fonti dati estemporanee (file Excel), da utilizzare fino al momento in cui le tabelle fossero state ultimate. In linea con questa scelta, sono stati realizzati i file Excel descritti di seguito.

- Grp e Grp 30##YYYYMM##HOLDING X.xlsx: è un file inerente ai GRP e presenta nel titolo il nome della holding cui i dati sono riferiti (al posto di “HOLDING X”) e la coppia mese-anno in cui i dati sono stati calcolati (al posto di “YYYYMM”). Il file contiene i valori di GRP e GRP equivalente (un valore di GRP calcolato rispetto ad un intervallo temporale di 30 secondi) declinati per anno, mese, target, mezzo e concessionario. Nello specifico, il concessionario coincide sempre con Rai, mentre il mezzo può variare tra “TV Generalista” e “TV Specializzata”.
- Target Clienti Accordi Quadro.xlsx: questo file contiene un elenco che indica, per ogni holding, il target di riferimento corrispondente. Tale informazione è utile sempre in funzione del GRP perché, come è possibile capire dalla descrizione del file precedente, ogni rilevamento del valore di GRP si riferisce ad un certo target.
- TV_CPGTREND##YYYYMM.xlsx: il file, che nel titolo presenta la coppia mese-anno in cui i valori sono stati calcolati (al posto di “YYYYMM”), contiene i dati su investimento netto e GRP equivalente relativi ai target “Adulti 25-54” e “Adulti 35-64”. Questi dati sono declinati per concessionario, mezzo, anno e mese e sono utili al calcolo dei valori di CPG (Costo Per GRP), rispetto ai due target appena citati. Il file ha come unica finalità quella di consentire il calcolo dei valori di benchmark per il CPG, in quanto tutti gli altri valori di CPG sono calcolabili in ogni momento, a partire dai dati sugli investimenti (contenuti nel DWH) e dai dati sul GRP (contenuti nei file Excel “Grp e Grp 30##YYYYMM##HOLDING X.xlsx”).
- VIEWABILITY CLIENTE SETTORE##201706.xlsx: è un file inerente alla viewability e presenta nel titolo la coppia mese-anno cui risale il rilevamento delle informazioni e cioè giugno 2017. In questo documento sono contenuti i valori di “Gross Impressions”, “Measured Impressions”, “Measured Views”, declinati per anno, mese, cliente e settore, che possono essere utilizzati per calcolare il valore di viewability, ottenibile come il rapporto tra measured views e measured impressions.
- Decodifica Clienti Viewability.xlsx: i clienti elencati nel file “VIEWABILITY CLIENTE SETTORE##201706.xlsx” non corrispondono, da un punto di vista anagrafico, a quelli presenti nei database aziendali e questo perché i dati sulla viewability non vengono raccolti da Rai Pubblicità, ma da ComScore, una società che si occupa di effettuare misurazioni sul web in ottica marketing. Questo file Excel ha come obiettivo quello di creare una corrispondenza tra clienti Rai Pubblicità e clienti ComScore, così da ottenere una connessione tra le informazioni esposte nel file e quelle presente all’interno del datawarehouse.

Anche se, sulla base di una prima analisi, era stata riscontrata l’assenza delle informazioni sulle posizioni pregiate, in un secondo momento, si è scoperto che tali informazioni erano presenti sotto l’istanza DB Commerciale e, semplicemente, non erano state ribaltate sotto l’istanza DWH. Per questa ragione, considerando anche il fatto che il DB Commerciale è un’istanza accessibile da Qlik, non è stato più necessario creare un file Excel dedicato alle posizioni pregiate.

3.4 Creazione delle viste predisposte per Qlik

Al fine di semplificare il lavoro dell'ETL 01 centralizzato, si è pensato di “standardizzare” i dati presenti all'interno dei database aziendali Oracle, affinché il primo livello ETL possa utilizzare, come fonti, degli oggetti creati appositamente, anziché sfruttare direttamente le tabelle originali. Più precisamente, si è deciso di creare, sia all'interno dell'istanza DWH che in quella DB Commerciale, delle viste che riuscissero a combinare i dati contenuti nelle tabelle da caricare in Qlik. L'introduzione di queste viste è stata pensata, principalmente, per evitare che una stessa risorsa fosse utilizzata come sorgente sia per l'ambiente Qlik che per lo svolgimento di altre operazioni aziendali, inoltre, questa soluzione consente, ancora prima di entrare in Qlik, di dettare le linee guida per la creazione della nuvola dati delle applicazioni, specificando quali sono le viste dei fatti (viste principali) e quali quelle anagrafiche e dimensionali (viste secondarie).

La creazione delle nuove viste è stata condotta utilizzando SQL Developer ed il processo di “standardizzazione” ha riguardato soprattutto, oltre alla struttura che ogni oggetto deve avere, la nomenclatura da utilizzare, sia per quello che riguarda le viste che per i campi in esse contenuti. Per quanto concerne il nome da assegnare alle viste, si è scelto di introdurre i seguenti prefissi:

- “VAN_” per le viste anagrafiche, ossia quelle viste che servono semplicemente ad associare codici e descrizioni.
- “VDM_” per le viste dimensionali, ossia quelle viste che combinano tra loro due o più anagrafiche che si riferiscono ad uno stesso ambito, ma si trovano a livelli di aggregazione differenti.
- “VDT_” per le viste contenenti i dati, ossia le viste principali, che espongono grandezze quantitative (misure), declinate su varie dimensioni.
- “VTM_” per le viste contenenti dati relativi alla dimensione temporale.
- “VPF_” per le viste contenenti i dati “di portafoglio”, ossia i dati che riguardano agenzie, venditori, etc.

Per quanto riguarda i campi, invece, i prefissi ed i suffissi da utilizzare sono i seguenti:

- “ID_” per il campo chiave, qualora questo sia presente in una tabella. Solitamente il campo chiave nasce come combinazione dei codici di due o più anagrafiche e serve ad identificare in modo univoco i record di una tabella.
- “ORD_” per il campo utile all'ordinamento dei record all'interno di una tabella.
- “IMP_” per i campi contenenti degli importi.
- “NR_” per i campi contenenti dei valori numerici.
- “DATA_” per i campi contenenti una data.
- “DATAUPD_” per i campi contenenti le date di update.
- “FL_” per i campi flag, contenenti un valore booleano.
- “_COM” per i campi contenenti valori commerciali.
- “_NLS” per i campi contenenti valori forniti dalla società Nielsen.
- “_RIALL” per i campi contenenti valori che hanno subito il processo di riallineamento.
- “_SA” per i campi utili alla gestione degli accessi, infatti, “SA” sta per “Section Access”.

Nella tabella riportata di seguito sono elencate tutte le viste create per le app “AQC” e “Flow”, mettendo in evidenza i singoli campi che le compongono e le tabelle originali dalle quali tali viste prendono spunto.

Vista	Tabella originale	Campo
VDT_INV_PUBBLICITARI	NLS_ST_MEZZI, NLS_RELAZ_MEZZI_TP_PRODOTTO, VI_CC_CFG_RELAZ_DGC_NODI_NL, NLS_ST_RELAZ_RETI, CC_MEZZI, VDM_PRODOTTO_PUBBLICITARIO, CC_DATI_MAGAZZINO	ID_INVESTIMENTO
		ID_GRP
		ID_PRODOTTOEDITORIALE
		ID_PRODOTTOPUBBLICITARIO
		ID_TIPI_SOTTOTIPI_PUBBLICITA
		ID_ATTUAZIONEVENDITA
		ID_MERCEOLOGIA
		ID_TIPIPIANO_TIPISCONTO
		ID_CALENDARIO_COM
		ID_EVENTO
		ID_INVESTIMENTO_FATTURA
		ID_INV_TRIPLACONT_REALTIME
		COD_MEDIA
		COD_CLIENTE_COM
		COD_MEZZO
		COD_JOB
		COD_PIANO
		COD_VERSIONE
		COD_ORIGINERICAVO
		COD_CONDIZIONEPAIAMENTO
		COD_CLIENTE
		COD_COMMITTENTE
		COD_COMMITTENTE_COM
		COD_AGENZIA
		COD_HOLDING
		COD_SOGGETTO
		COD_CLIENTEORIGINARIO
		COD_SOGGETTOORIGINARIO
		NR_FORMATO
		COD_TIPOPDUZIONE
		COD_STATOPRODOTTOSIPRALAB
		COD_TARGETMACRO
		COD_TARGETMICRO
COD_TARGETRIFERIMENTO		
COD_DETTFASCIAEFFETTIVA		
COD_DETTDASCIAPREVISTA		
COD_DETTFASCIALISTINO		
DATA_TRASMISSIONE		
FL_TRASMESSO		

		COD_TIPOPIANO
		COD_TIPOCONTRATTO
		COD_TIPOSCONTO
		COD_MAGAZZINO
		COD_DGC
		COD_DGCPROVVIGIONI
		COD_STAOVENDITA
		COD_STATOCONTABILE
		COD_RAGGMEZZO_NLS
		COD_MEZZO_NLS
		COD_RETE_NLS
		IMP_LORDOLISTINO
		IMP_SCTOTALE
		IMP_NETTO
		IMP_NETTOLORDIZZATO
		IMP_LORDOCOMMERCIALE
		IMP_SCCOMMERCIALE
		IMP_SCCOMMERCIALEFINANZIARIO
		IMP_SCCOMMERCIALEOMAGGIO
		IMP_SCSANATORIA
		IMP_SCRECUPEROTOTALE
		IMP_SCRECUPEROOTTIMIZZAZIONE
		IMP_SCRECUPEROINTEGRAZIONE
		NR_PASSAGGI
		NR_SECONDS
		NR_IMPRESSIONVENDUTE
		IMP_NETTOIMPRESSIONVENDUTE
		NR_IMPRESSIONEFFEROGATE
		IMP_NETTOIMPRESSIONEFFEROGATE
		COD_SETTOREDOMINANTE
		COD_CACQ_COM
		COD_LISTINO
		COD_MACROFORMATOWEB
		COD_MODALITAFRUIZIONEWEB
		COD_MODALITAVENDITACINEMA
		COD_TIPOSALACINEMA
		COD_CONDPARTICOLAREVENDITA
		COD_DGCCALCOLATO
		COD_TIPOCONTRATTOCALCOLATO

VDT_INV_MERCATO	VM_CC_INV_MERCATO	ANNO
		MESE
		SETTIMANA
		COD_RAGGCONCESSIONARIO_NLS
		COD_CONCESSIONARIO
		COD_RAGGMEZZO_NLS
		COD_MEZZO
		COD_TIPOLOGIA
		COD_TIPOCOMUNICAZIONE
		COD_RETE_NLS
		COD_RAGGRMEZZO_RAIPUBBL
		COD_MEZZO_RAIPUBBL
		COD_REGIONE_NLS
		COD_CLIENTE
		COD_SETTORE_NLS
		COD_SETTOREDOMINANTE
		SETTOREDOMINANTE
		COD_STATOCLIENTE
		COD_STATOPORTAFOGLIO
		COD_HOLDING_NLS
		COD_CLIENTE_NLS
		COD_HOLDING
		COD_CLIENTE_COM
		COD_GRUPPOCMEDIA_RAIPUBBL
		COD_CENTROMEDIA_RAIPUBBL
		COD_AGENZIA_COM
		COD_AREA
		COD_GRUPPO
		COD_VENDITORECLIENTE
		COD_AREA_VENDITOREAGENZIA
		COD_GRUPPO_VENDITOREAGENZIA
		COD_VENDITOREAGENZIA
		COD_AREA_VENDITOREPRODOTTO
		COD_GRUPPO_VENDITOREPRODOTTO
		COD_VENDITOREPRODOTTO
		COD_AREA_VENDITOREPRODOTTOAGZ
		COD_GROUP_VENDITOREPRODOTTOAGZ
		COD_VENDITOREPRODOTTOAGENZIA
		COD_TIPODATO
		COD_STATOVENDITA
STATOVENDITA		
ORD_STATOVENDITA		
IMP_LORDO_NLS		

		IMP_NETTO_NLS
		IMP_NETTOMARKETING_NLS
		COD_STATO_PRES_RAIPUBBL
		IMP_LORDO_RAIPUBBL
		IMP_NETTO_RAIPUBBL
		PERC_RIP_LRD_CLIENTE_NIELSEN
		PERC_RIP_LRD_CLIENTE_RAIPUBBL
		IMP_LORDO
		IMP_NETTO
		IMP_LORDO_CONCORRENZA
		IMP_NETTO_CONCORRENZA
		FL_AZIENDA_GENERICA
		COD_TIPOPUBBLICITA
		FL_CARICAMENTO
		STATO
		ID_INTERLOCUTORE_PF
		IMP_SCONTO_NLS
		NR_SECONDI
		ANNUNCI_RAIPUBBL
		SECONDI
		SCHERMI
		NR_IMPRESSIONS
		INVII
		NR_ANNUNCI
		IMP_NETTO_MAGAZZINO
		IMP_RICAVO
		IMP_SANATORIA
		IMP_RECUPERO
		IMP_SCONTO_TOTALE
		COD_MEZZO_RP
		INDIVIDUI_GRP
		ADULTI_25_44_GRP
		ADULTI_25_54_GRP
		RA_GRP
		RA_25_54_GRP
		UOMINI_15_GRP
		DONNE_15_GRP
		KIDS_4_14_GRP
		INDIVIDUI_GRP_ARI
		ADULTI_25_44_GRP_ARI
		ADULTI_25_54_GRP_ARI
		ADULTI_35_64_GRP_ARI
		RA_GRP_ARI
		RA_25_54_GRP_ARI

		UOMINI_15_GRP_ARI
		DONNE_15_GRP_ARI
		KIDS_4_14_GRP_ARI
		ANNUNCI_ARI
		COD_ABBINAMENTO_AGB_NIELSEN

VAN_AGENZIE_COMMERCIALI	VI_CC_AGENZIE_COMM	COD_AGENZIA_COM
		AGENZIA_COM
VAN_AREE	CC_AREE	COD_AREA
		AREA
		ORD_AREA
VAN_CATEGORIE_PUBBLICITARIE	CC_CATEGORIA_PRODOTTO	COD_CATEGORIAPUBBLICITARIA
		CATEGORIAPUBBLICITARIA
		ORD_CATEGORIAPUBBLICITARIA
VAN_CLASSI_INTERVENTO	CLASSE_INTERVENTO	COD_CLASSEINTERVENTO
		CLASSEINTERVENTO
VAN_CLIENTI_COMMERCIALI	CC_INTERL_U, CC_ANAG_CLASSIFICAZIONE	COD_CLIENTE_COM
		CLIENTE_COM
VAN_CLIENTI_NIELSEN	NLS_ST_AZIENDA	COD_CLIENTE_NLS
		CLIENTE_NLS
		COD_CLIENTE_NLS_NL
VAN_EDITORI	CC_EDITORI	COD_EDITORE
		EDITORE
		ORD_EDITORE
VAN_EVENTI	V_CC_MANIFESTAZIONE	COD_EVENTO
		EVENTO
		DATA_INIZIOEVENTO
		DATA_FINEEVENTO
		ORD_EVENTO
		COD_TIPOEVENTO
VAN_HOLDING	CC_INTERL_U, CC_RAGGRUPPAMENTO_U, VAN_CLIENTI_COMMERCIALI	COD_HOLDING
		HOLDING
VAN_MEDIA_PUBBLICITARI	CC_MEDIA	COD_MEDIA
		MEDIA
		ORD_MEDIA
VAN_MEZZI_PUBBLICITARI	CC_MEZZI	COD_MEZZO
		MEZZO
		ORD_MEZZO
		COD_MEDIA
VAN_MEZZI_PUBBLICITARI_NIELSEN	NLS_ST_MEZZI	COD_MEZZO_NLS
		MEZZO_NLS
		ORD_MEZZO_NLS
		COD_RAGGMEZZO_NLS

VAN_PRODOTTI_PUBBLICITARI	CC_MAGAZZINI, CC_AGGR_STND_TIPI_VEND	AST_CODICE
		AST_DESCRIZIONE
		MAG_MEZ_CODICE
		AST_ORDINAMENTO
VAN_RAGG_CONCESSIONARI_NIELSEN	NLS_ST_RAGG_CONCESSIONARI	COD_RAGGCONCESSIONARIO_NLS
		RAGGCONCESSIONARIO_NLS
		ORD_RAGGCONCESSIONARIO_NLS
VAN_RAGG_MEZZI_NIELSEN	NLS_ST_RAGG_MEZZI	COD_RAGGMEZZO_NLS
		RAGGMEZZO_NLS
		ORD_RAGGMEZZO_NLS
VAN_RETI	CC_RETI	COD_RETE
		RETE
		ORD_RETE
		COD_MEZZO
VAN_RETI_NIELSEN	NLS_ST_RETI	COD_RETE_NLS
		RETE_NLS
		ORD_RETE_NLS
		COD_MEZZO_NLS
VAN_SETTORI_NIELSEN	V_CC_NIELSEN_SETTORI	COD_SETTORE_NLS
		SETTORE_NLS
		FL_VAL_SETTORE_NLS
VAN_TESTATE_EDITORIALI	CC_TESTATE_EDITORIALI	COD_TESTATAEDITORIALE
		TESTATAEDITORIALE
		ORD_TESTATAEDITORIALE
		COD_EDITORE
VAN_TIPI_EVENTO	V_CC_MANIFESTAZIONE	COD_TIPOEVENTO
		TIPOEVENTO
		ORD_TIPOEVENTO
VAN_TIPI_PUBBLICITA	CC_TIPO_VEND_MEZZO	COD_TIPOPUBBLICITA
		TIPOPUBBLICITA
		COD_PRODOTTOPUBBLICITARIO
VAN_VENDITORI	VI_CC_VENDITORI	COD_VENDITORE
		VENDITORE
		COD_TIPOINTERLOCUTORE
VDM_EVENTO	VAN_TIPI_EVENTO, VAN_EVENTI	ID_EVENTO
		COD_EVENTO
		EVENTO
		DATA_INIZIOEVENTO
		DATA_FINEEVENTO
		ORD_EVENTO
		COD_TIPOEVENTO
		TIPOEVENTO
		ORD_TIPOEVENTO

VDM_MARCHE_CLIENTI_NIELSEN	NLS_ST_PRODOTTI_INV, VAN_CLIENTI_NIELSEN	COD_MARCA_NLS
		MARCA_NLS
		COD_TIPOMARCA_NLS
		COD_CLIENTE_NLS
		CLIENTE_NLS
VDM_MEZZI_RP_MEZZI_NIELSEN	NLS_ST_MEZZI, NLS_RELAZ_MEZZI_TP_PRODOTTO, VI_CC_CFG_RELAZ_DGC_NODI_NL, VAN_MEZZI_PUBBLICITARI, VAN_MEZZI_PUBBLICITARI_NIELSEN, VAN_RAGG_MEZZI_NIELSEN	COD_MEZZO
		MEZZO
		COD_MEZZO_NLS
		MEZZO_NLS
		COD_RAGGMEZZO_NLS
VDM_PRODOTTO_PUBBLICITARIO	VAN_CATEGORIE_PUBBLICITARIE, VI_PC_RELAZ_DGC_SV, VAN_MEDIA_PUBBLICITARI, VAN_MEZZI_PUBBLICITARI, VAN_PRODOTTI_PUBBLICITARI, VAN_TIPI_PUBBLICITA	RAGGMEZZO_NLS
		ID_PRODOTTOPUBBLICITARIO
		COD_MEDIA
		MEDIA
		COD_MEZZO
		MEZZO
		COD_PRODOTTOPUBBLICITARIO
		PRODOTTOPUBBLICITARIO
		COD_TIPOPUBBLICITA
		TIPOPUBBLICITA
		COD_CATEGORIAPUBBLICITARIA
CATEGORIAPUBBLICITARIA		
VDM_PRODOTTO_EDITORIALE	VAN_EDITORI, VAN_TESTATE_EDITORIALI, VAN_RETI, CC_DATI_MAGAZZINO	TIPO_TRADIZ_INTERATT
		COD_TIPOPUBBLICITA_SA
		ID_PRODOTTOEDITORIALE
		COD_RETE
		RETE
		COD_EDITORE
VDM_RETI_RP_RETI_NIELSEN	VAN_RETI_NIELSEN, NLS_ST_RELAZ_RETI, VAN_RETI	EDITORE
		COD_TESTATAEDITORIALE
		TESTATAEDITORIALE
		ID_RETE
		COD_RETE
		RETE
		ORD_RETE
VDM_SOGGETTI_CLIENTI	VI_CC_SOGGETTI, VAN_CLIENTI_COMMERCIALI	COD_RETE_NLS
		RETE_NLS
		ORD_RETE_NLS
		COD_CLIENTE_COM
		CLIENTE_COM
		COD_SOGGETTO
		SOGGETTO
		COD_CLIENTE_SA

VPF_AGENZIE	V_AN_PORT_SOG_OF_CM	ID_PORTAFOGLIOAGENZIE
		COD_CLIENTE_COM
		COD_SOGGETTO
		COD_AGENZIA_COM
		COD_MEZZO
		DATA_INIZIO
		DATA_FINE
VPF_VENDITORI_CLIENTI	CC_PORTAFOGLIO_ATTIVO, GEST_COM	COD_VENDITORECLIENTE
		COD_CLIENTE_COM
		COD_AREA
		COD_CLASSEINTERVENTO
		COD_MEZZO
VPF_VENDITORI_AGENZIE	CC_PORTAFOGLIO_ATTIVO, GEST_COMM	COD_VENDITOREAGENZIA
		COD_AGENZIA_COM
		COD_AREA
		COD_CLASSEINTERVENTO
		COD_MEZZO
VDT_INV_PUBBL_POS_BREAK_TV	SPOTS_TV, V_RT_RETI	CHIAVE_COMUNICATO
		FL_POSIZIONE_PRIVILEGIATA
VDT_INV_PUBBL_POS_BREAK_RADIO	SPOTS_RN, V_RT_RETI	CHIAVE_COMUNICATO
		FL_POSIZIONE_PRIVILEGIATA

Mentre la funzione delle viste anagrafiche e dimensionali è facilmente intuibile, le viste dei fatti, ossia “VDT_INV_PUBBLICITARI” e “VDT_INV_MERCATO”, necessitano di un approfondimento. La “VDT_INV_PUBBLICITARI” ospita i dati relativi agli investimenti effettuati su prodotti Rai Pubblicità, mentre la “VDT_INV_MERCATO” contiene le informazioni fornite da Nielsen, relative agli investimenti della clientela sui prodotti dei principali competitors di Rai. Entrambe le viste, poi, presentano una serie di “valori chiave” utili ad agganciare le viste anagrafiche e dimensionali (VAN e VDM) e a realizzare il mapping tra le anagrafiche interne Rai e quelle utilizzate da Nielsen. Senza il mapping risulterebbe impossibile creare una corrispondenza tra le informazioni e di conseguenza, una volta in Qlik, risulterebbe molto più complessa la realizzazione della nuvola dati.

Alle viste elencate in tabella, va aggiunta la vista “VDT_DATI_NIELSEN” che ha come sorgente la tabella “VM_CC_DATI_NIELSEN” e contiene le stesse informazioni della “VDT_INV_MERCATO”. L’unica differenza tra questi due oggetti sta nel fatto che all’interno della “VDT_DATI_NIELSEN” gli investimenti sono esplicitati per marca e per settimana, scendendo, perciò, ad un livello di aggregazione più basso rispetto alla “VDT_INV_MERCATO” ed è proprio questo il motivo per cui tale vista viene utilizzata come sorgente dati per l’applicazione “Flow”, la quale prevede l’analisi dell’andamento settimanale degli investimenti della clientela, dettagliati per marca.

In generale queste viste, essendo state create ancora prima dell’ETL 01 centralizzato, non saranno di aiuto alle sole app “AQC” e “Flow”, ma verranno utilizzate a supporto di più applicazioni; le viste dei fatti “VDT_INV_PUBBLICITARI”, “VDT_INV_MERCATO” e “VDT_DATI_NIELSEN” ad esempio, ospitano informazioni che esulano dalle app cui la tesi fa riferimento, ma tali informazioni potranno essere “scartate” in un secondo momento, quando verrà eseguito il caricamento dell’ETL 01 di app.

3.5 Configurazione del framework ETL

Come è stato ampiamente spiegato nel paragrafo relativo al funzionamento del sistema di BI aziendale, gli ETL devono essere opportunamente configurati per poter veicolare le informazioni dai database Oracle alle applicazioni Qlik Sense. Nel caso delle app “AQC” e “Flow”, non trattandosi dei primi progetti sviluppati dal team di BI, la configurazione ha riguardato soprattutto ETL 01 ed ETL 02 di app, mentre gli ETL centralizzati risultavano già predisposti e, di conseguenza, il lavoro operato su di essi è stato minimo. Di seguito saranno illustrate le modifiche apportate alle tabelle di configurazione:

- ETL_GLOBAL_VARIABLES: le variabili necessarie agli ETL centralizzati erano già presenti in tabella, mentre è stato necessario aggiungere quelle contenenti i path relativi al nuovo progetto, utili agli ETL di app (path in cui leggere/scrivere i QVD prodotti dagli ETL di app e path in cui leggere i file Excel sorgenti per l’ETL 01 di app).
- ETL_01_CONNECTION: le connessioni necessarie alle app “AQC” e “Flow” (connessione al DWH e al DB Commerciale) risultavano già presenti in tabella.
- ETL_01_TABLETOEXTRACT: per quanto riguarda l’ETL 01 centralizzato, è stato necessario aggiungere, ponendo nel campo “PROJECTID” il valore 1, una riga per ogni nuova vista creata sul database Oracle. Per configurare l’ETL 01 di app, invece, è stata aggiunta una riga per ogni nuovo QVD prodotto dall’ETL 02 centralizzato e una riga per ciascun file Excel realizzato appositamente per queste app. Bisogna sottolineare che, a differenza delle righe propedeutiche all’ETL 01 centralizzato, quelle utili all’ETL 01 di app presentano nel campo “PROJECTID” il valore “1002” (valore scelto come identificativo delle app “AQC” e “Flow”).
- ETL_01_QUERY: nel caso dell’ETL 01 centralizzato, non sono state necessarie modifiche, in quanto a questo livello le tabelle vanno caricate per intero senza specificare i singoli campi. Facendo riferimento all’ETL 01 di app, invece, è stata aggiunta una riga per ciascun campo da caricare che fosse utile alle app “AQC” e “Flow”. In questo modo, i QVD prodotti dall’ETL 01 di app contengono solo un sottoinsieme dei campi caricati inizialmente e riducono la quantità di informazioni da gestire nei livelli successivi. Anche in questo caso, il valore di “PROJECTID” utilizzato per i record è stato “1002”.
- ETL_01_VARIABLES: come per la tabella ETL_01_QUERY, anche in questo caso non sono state apportate modifiche in relazione all’ETL 01 centralizzato, mentre sono state aggiunte delle variabili, utili all’ETL 01 di app, appositamente create per le app “AQC” e “Flow”.
- ETL_02_INDEX: al fine di configurare l’ETL 02 centralizzato, è stato necessario aggiungere una riga per ogni nuovo QVD generato dall’ETL 01 centralizzato. Per quanto riguarda l’ETL 02 di app, invece, sono stati aggiunti i record relativi ai QVD, prodotti dall’ETL 01 di app, dedicati alle app “AQC” e “Flow”.
- ETL_02_TABLES: le uniche modifiche apportate a questa tabella hanno riguardato l’ETL 02 di app; nello specifico, è stato aggiunto, per ciascun campo presente all’interno QVD prodotti dall’ETL 01 di app, un record indicante le trasformazioni da operare su quel campo.
- QLIK_CATALOGO_INFORMAZIONI: in questa tabella è stata aggiunta una riga per ognuno dei nuovi campi caricati nell’ambito delle applicazioni “AQC” e “Flow”. Le righe contenute nel catalogo informativo permettono di associare al nome tecnico di ogni campo la descrizione da visualizzare in fase di presentazione del dato.

Come già spiegato in precedenza, il framework ETL è realizzato mediante applicazioni Qlik Sense, il cui script di caricamento viene generato in modo dinamico, in base al contenuto delle tabelle di configurazione. Esistono, però, delle operazioni di trasformazione sui dati che non possono essere eseguite in modo dinamico e vanno implementate mediante l'inserimento di codice aggiuntivo all'interno delle applicazioni. Nel sistema Rai Pubblicità, le app ETL 01 centralizzato, ETL 02 centralizzato, ETL 01 di app ed ETL 02 di app presentano, nell'area dedicata al caricamento, una sezione aggiuntiva, chiamata "Custom", predisposta per ospitare eventuali script aggiuntivi, utili al processo di elaborazione delle informazioni. Nel caso specifico delle app "AQC" e "Flow" sono state utilizzate sia la sezione custom dell'app ETL 02 centralizzato, che quella dell'ETL 02 di app. Nella sezione "Custom" dell'ETL 02 centralizzato è stato inserito uno script per mettere in join la tabella relativa agli investimenti (VDT_INV_PUBBLICITARI) con le tabelle, estratte dal DB Commerciale, contenenti le informazioni sulle posizioni pregiate (VDT_INV_PUBBL_POS_BREAK_TV e VDT_INV_PUBBL_POS_BREAK_RADIO). All'interno della sezione "Custom" dell'ETL 02 di app, invece, il codice inserito risulta di gran lunga più corposo, in quanto consente, oltre alla creazione delle tabelle di link, utili alla strutturazione del modello dati, la creazione di una tabella per definire la dimensione temporale (Calendario), l'esecuzione delle operazioni di group by, join e mapping e, infine, il calcolo del settore prevalente, corrispondente ad ogni holding. Tutte le queste operazioni, oltre a comportare una riduzione della quantità di dati generati per l'app di front-end, tracciano la strada per la costruzione del modello dati, al di sopra del quale sarà costruita l'applicazione finale. Tale modello verrà generato in maniera automatica, sfruttando la capacità, presente in Qlik, di creare relazioni tra campi omonimi, contenuti in tabelle differenti.

Potrebbe sembrare strano il fatto di aver illustrato la configurazione di un'unica coppia di ETL di app, sebbene sia stata presa la decisione di articolare l'applicativo richiesto in due applicazioni Qlik Sense differenti ("AQC" e "Flow"). La ragione dell'assenza di un ulteriore coppia di ETL di app risiede nel fatto che le informazioni necessarie all'app "Flow" costituiscono un sottoinsieme delle informazioni necessarie all'app "AQC" e, di conseguenza, entrambe le applicazioni possono attingere ai QVD generati dall'unico ETL 02 di app presente. Ovviamente, saranno due le app di front-end e ciascuna costruirà al proprio interno un modello dati differente, sebbene le fonti siano comuni.

3.6 Cenni sulla creazione del modello dati

Come già sottolineato nella sezione introduttiva del presente elaborato, la progettazione dei modelli dati sottesi alle applicazioni "AQC" e "Flow" è stata oggetto di un lavoro di tesi svolto parallelamente a questo e, pertanto, nel paragrafo corrente ci si limiterà ad analizzare la struttura di tali modelli, senza spiegare quali decisioni siano state prese per giungere al risultato finale. Per primo sarà analizzato il modello dati dell'app "AQC", in quanto risulta molto esteso ed articolato, dopodiché si passerà all'analisi del modello dati su cui si erge l'app "Flow" che, come si vedrà, è notevolmente meno complesso.

Prima di procedere è necessario illustrare, anche in base alla definizione di "rappresentazione multidimensionale delle informazioni", fornita nel capitolo 1, cosa si intenda in Qlik con i termini "tabella dei fatti", "tabella delle dimensioni" e "tabella di link".

- Tabella dei fatti (fact table): nel capitolo 1 il "fatto" è stato descritto come un oggetto utile a modellare un qualsiasi evento che sia rilevante rispetto al contesto aziendale; di conseguenza, la fact table è l'oggetto che viene utilizzato in Qlik per la memorizzazione dei dati necessari all'analisi di un evento. La tabella dei fatti è l'elemento principe del data

model e contiene un insieme di valori quantitativi (“misure”) che, combinati con le grandezze dimensionali, consentono di “fotografare” un evento aziendale da molteplici prospettive. Generalmente la tabella dei fatti contiene un numero molto elevato di record ed è quindi consigliato limitare il numero dei campi dimensionali contenuti al suo interno, onde evitare un degrado delle prestazioni in fase di interrogazione.

- Tabella delle dimensioni: la tabella delle dimensioni è solitamente collegata a quella dei fatti mediante un campo chiave e contiene i dati relativi a una delle prospettive utilizzate per la declinazione e l’analisi delle misure. Solitamente le tabelle dimensionali contengono molti meno record rispetto alla tabella dei fatti, ma possono contenere una grande quantità di campi, specialmente quando fanno riferimento ad una struttura gerarchica (anno, semestre, mese, settimana).
- Tabella di link (link table): questo tipo di tabella non ha nulla a che fare con il concetto di multidimensionalità delle informazioni, ma costituisce un elemento che può tornare utile in fase di costruzione del modello dati, in quanto permette il collegamento di più tabelle dei fatti. La link table contiene un insieme di campi che due o più fact table hanno in comune e si collega a queste ultime grazie ad una chiave, appositamente creata, data dalla concatenazione di tali campi. Tale soluzione ha come obiettivo quello di lasciare sulle fact table soltanto i campi chiave, spostando sulle tabelle di link sia i record comuni, che quelli presenti solo in una delle tabelle dei fatti.

Il data model relativo all’app “AQC”, come è possibile evincere dalla figura sottostante, consta di dodici tabelle dimensionali (quelle poste sulla destra), cinque tabelle dei fatti (quelle poste al centro) e tre tabelle di link (quelle poste sulla sinistra) ed è strutturato secondo quello che viene detto “star schema”, ossia un modello denormalizzato in cui è presente una fact table centrale, collegata direttamente ad un gran numero di tabelle dimensionali.

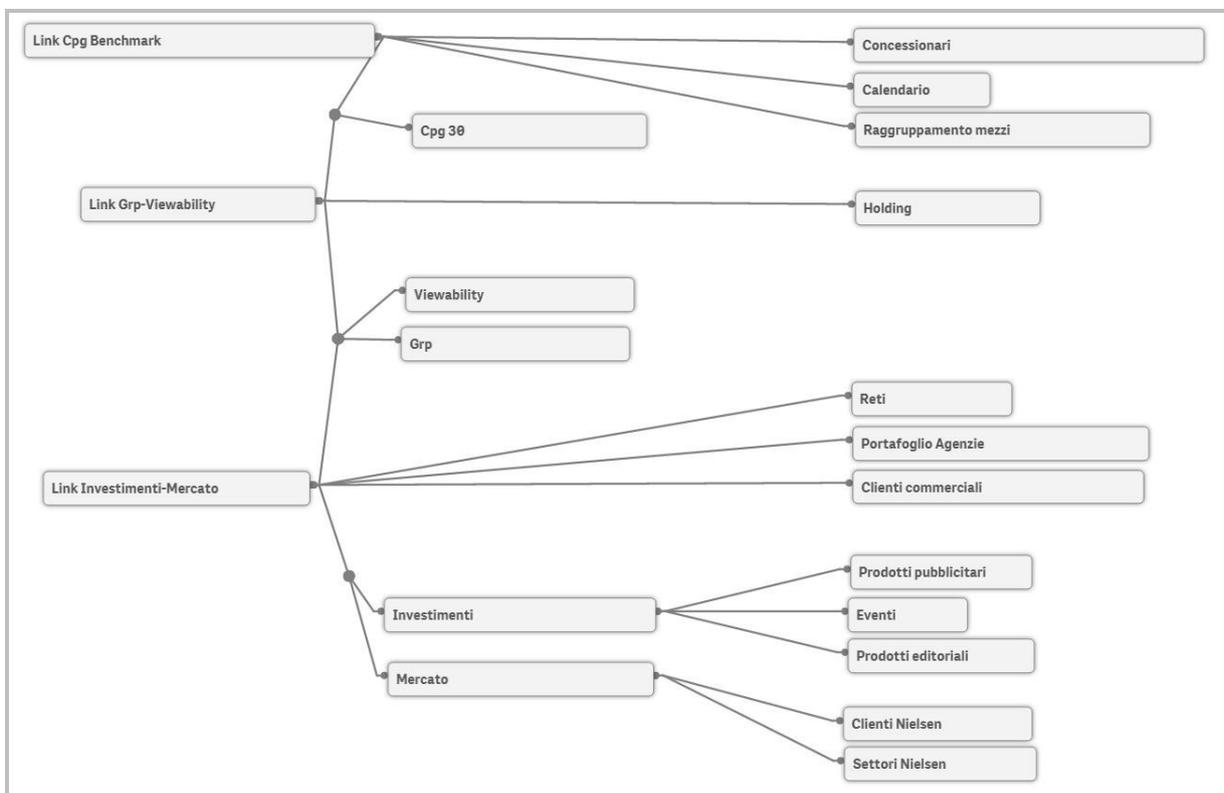


Figura 11 - Struttura del data model costruito per l'app "AQC"

La struttura a stella permette di ottenere un compromesso tra consumo di memoria RAM e prestazioni, in quanto la presenza di un unico livello dimensionale porta, da un lato, all'alleggerimento della tabella dei fatti e, dall'altro, alla riduzione del tempo di risposta alle query.

Di seguito saranno illustrate, una per una, tutte le tabelle che compongono il modello dati dell'app "AQC", facendo attenzione al modo in cui sono state collegate tra loro.

- Concessionari: è una tabella dimensionale che ospita le informazioni caricate a partire dalla vista materializzata VAN_RAGG_CONCESSIONARI_NIELSEN. I soli campi esposti, oltre al "Codice Raggruppamento Concessionario Nielsen", che è un campo univoco ed è necessario ad agganciare la dimensione alla link table "Link Cpg Benchmark", sono un campo descrittivo ed un campo utile all'ordinamento.
- Calendario: è probabilmente la tabella dimensionale più importante ed è popolata a partire dalle informazioni temporali (semestre, anno, mese, settimana, etc.) raccolte analizzando tutte le tabelle dei fatti. Nel dettaglio, sono state individuate la prima e l'ultima data cui fossero associate delle informazioni, dopodiché sono stati generati tutti i possibili valori di calendario compresi in questo arco di tempo. Anche la tabella Calendario è agganciata alla link table "Link Cpg Benchmark" mediante il campo "ID Calendario", ossia la data in formato gg/mm/aaaa.
- Raggruppamento mezzi: questa tabella dimensionale è stata realizzata a partire dalle informazioni contenute nella vista VAN_RAGG_MEZZI_NIELSEN ed espone, al pari della tabella "Concessionari", il "Codice Raggruppamento Mezzo Nielsen", un campo descrittivo ed un campo finalizzato ad operazioni di ordinamento. Il campo "Codice Raggruppamento Mezzo Nielsen" è utilizzato per agganciare la presente tabella alla link table "Link Cpg Benchmark".
- Holding: la presente tabella contiene le informazioni anagrafiche relative alle holding o ai singoli clienti, nel caso in cui costoro non risultino afferenti ad alcuna holding. La fonte utilizzata per il popolamento di questa tabella è la VAN_HOLDING, ma ai dati estratti dalla vista sono aggiunti quelli su settore prevalente, target di riferimento e possibilità di stipula di un accordo quadro. Il campo principale è sicuramente il "Codice Holding", necessario per agganciare la tabella alla link table "Link Grp-Viewability".
- Reti: questa tabella contiene l'anagrafica delle reti, sia Nielsen che Rai Pubblicità, estratta dalla vista VDM_RETI_RP_RETI_NIELSEN. In generale tutti gli oggetti del presente data model si basano su valori di rete che fanno riferimento all'anagrafica Nielsen, sostituendo, in alcuni casi, i valori di rete forniti da Rai Pubblicità con il corrispondente valore Nielsen. Il campo utilizzato per agganciare la presente tabella alla link table "Link Investimenti-Mercato" è il "Codice Rete Nielsen".
- Portafoglio Agenzie: a differenza delle tabelle analizzate fino ad ora, la tabella "Portafoglio Agenzie" non preleva i dati da una vista anagrafica (VAN) ma dalle due viste dimensionali VPF_AGENZIE e VPF_VENDITORI_AGENZIE, le quali convogliano tutte le informazioni necessarie a definire un'agenzia di vendita (nome, mezzi, venditori, aree venditori, classi intervento, etc.). Sebbene la tabella metta assieme molteplici dimensioni, il campo utilizzato per l'aggancio con la link table "Link Investimenti-Mercato" non è dato da una concatenazione di più colonne, ma dal campo "ID Portafoglio Agenzie", ossia il codice cliente.

- Clienti Commerciali: questa tabella combina le informazioni relative ai clienti, prelevate dalla vista materializzata VAN_CLIENTI_COMMERCIALI, con quelle relative ai venditori, prelevate invece dalla vista VPF_VENDITORI_CLIENTI. I venditori presi in considerazione sono quelli che colloquiano direttamente con il cliente e sono diversi dai “Venditori di agenzia”, inseriti nella “Portafoglio Agenzie”. La presente tabella si aggancia alla link table “Link Investimenti-Mercato” mediante il campo “Codice Cliente Commerciale”.
- Prodotti Pubblicitari: la presente tabella raccoglie informazioni riguardo i prodotti pubblicitari, ossia il media, il mezzo e la categoria pubblicitaria. I dati sono caricati a partire dalla vista materializzata VDM_PRODOTTO_PUBBLICITARIO ed il campo utilizzato per agganciare la tabella alla fact table “Investimenti” è “ID Prodotto Pubblicitario”.
- Eventi: la tabella eventi attinge dalla vista VDM_EVENTO e contiene le informazioni inerenti determinate manifestazioni. In particolare, se l’evento è “mondiali di calcio”, “europei di calcio”, “olimpiadi” o “festival di Sanremo”, allora il campo TIPO_EVENTO assume valore “Evento Speciale”. La chiave utilizzata per agganciare la tabella alla fact table “Investimenti” è “ID Evento”, ossia il codice dell’evento.
- Prodotti Editoriali: questa tabella carica i dati a partire dalla vista materializzata VDM_PRODOTTO_EDITORIALE e contiene le informazioni relative agli editori e alle testate editoriali. Il campo utilizzato per agganciare la tabella alla fact table “Investimenti” è “ID Prodotto Editoriale”.
- Clienti Nielsen: la tabella Clienti Nielsen ospita i campi codice e descrizione, utili a rappresentare i clienti secondo l’anagrafica Nielsen. Per il caricamento dei dati si attinge alla vista materializzata VAN_CLIENTI_NIELSEN ed il campo che consente l’aggancio con la fact table “Mercato” è il “Codice Cliente Nielsen”.
- Settori Nielsen: la presente tabella è utile a rappresentare i settori cui fanno riferimento le holding ed i clienti e segue l’anagrafica Nielsen. La vista materializzata dalla quale vengono caricate le informazioni è la VAN_SETTORI_NIELSEN ed il campo utilizzato per agganciare la tabella alla fact table “Mercato” è il “Codice Settore Nielsen”.
- Investimenti: tra le tabelle dei fatti, la tabella “Investimenti” è quella principale, in quanto ospita i dati inerenti al mondo Rai, originariamente contenuti all’interno della vista VDT_INV_PUBBLICITARI. Anche le informazioni contenute nelle due viste VDT_INV_PUBBL_POS_BREAK_TV e VDT_INV_PUBBL_POS_BREAK_RADIO, facenti riferimento alle posizioni pregiate, sono state inglobate nella tabella “Investimenti”, effettuando un’operazione di left join sul campo comune CHIAVE_COMUNICATO. La tabella risultante espone una serie di misure (“Netto Lordizzato”, “Lordo Listino”, “Sconto Cliente”, etc.) declinate rispetto alle dimensioni cliente commerciale, holding, raggruppamento mezzi, rete, evento, prodotto pubblicitario, prodotto editoriale, anno e mese.
- Mercato: la seconda tabella dei fatti in ordine di importanza è la “Mercato” che ospita i dati, forniti da Nielsen, inerenti agli investimenti effettuati dai competitors di Rai Pubblicità. La tabella “Mercato” attinge dalla vista VDT_INV_MERCATO e, analogamente alla tabella “Investimenti”, espone le misure “Netto Nielsen”, “Netto Marketing Nielsen” e “Netto Accordo Quadro Nielsen”, declinandole per le dimensioni cliente commerciale, cliente Nielsen, holding, raggruppamento concessionari, raggruppamento mezzi, rete, settore, tipo periodo, anno e mese.

- Grp: questa tabella è costruita a partire dalle fonti dati estemporanee (file Excel) relative ai dati collezionati da Rai Pubblicità sul Grp ed ospita i valori di “GRP” e “GRP Equivalente”, declinati per target, tipo target, holding, raggruppamento mezzi, raggruppamento concessionari, anno e mese. Bisogna sottolineare che, al fine di uniformare le informazioni su holding e raggruppamento mezzi presenti nel file Excel con quelle proprie delle anagrafiche Rai Pubblicità, è stato necessario effettuare una serie di operazioni di mapping all’interno di Qlik.
- Viewability: come la tabella “Grp”, anche questa tabella è popolata a partire dai valori caricati dalle fonti dati estemporanee relative al mondo web. Internamente, la tabella espone le informazioni raccolte da ComScore (“Gross Impressions”, “Measured Impressions” e “Measured Views”), declinate per cliente ComScore, settore ComScore, holding, raggruppamento concessionari, raggruppamento mezzi, anno e mese. Anche in questo caso le anagrafiche ComScore sono risultate incompatibili rispetto alle corrispondenti anagrafiche Rai ed è per questo motivo che sono state necessarie delle operazioni di mapping per uniformare i dati degli Excel al resto del modello.
- Cpg 30: questa tabella contiene le informazioni utili al calcolo del CPG 30 e attinge, come le due tabelle illustrate in precedenza, dalle fonti dati estemporanee predisposte per l’app. Le misure esposte sono “Grp 30” e “Netto per Cpg Benchmark” e sono declinate rispetto alle dimensioni target, raggruppamento concessionari, raggruppamento mezzi, anno e mese.
- Link Investimenti-Mercato: questa link table viene utilizzata per collegare le due tabelle dei fatti “Investimenti” e “Mercato” e per agganciare le tabelle dimensionali “Reti”, “Portafoglio Agenzie” e “Clienti Commerciali”, comuni ad entrambe le tabelle dei fatti. Mentre per le tabelle dimensionali è già stata indicata la tabella di aggancio, per quanto riguarda le tabelle dei fatti, bisogna sottolineare che queste vengono agganciate per mezzo della chiave composta “KEY_LINK_TABLE_INV_MER”, data dalla concatenazione dei campi comuni. Nello specifico, la chiave è composta dai valori di “Codice Cliente”, “Codice Raggruppamento Concessionari Nielsen”, “Codice Raggruppamento Mezzi Nielsen”, “Codice Rete Nielsen”, “Anno”, “Mese” e “Tipo Periodo Nielsen”, separati dal carattere ‘-’. All’interno della presente tabella è presente, inoltre, la chiave “KEY_LINK_TABLE_GRP_VW” che consente il collegamento con la link table “Link Grp-Viewability”.
- Link Grp-Viewability: questa seconda tabella di link è utile a connettere le tabelle dei fatti “Grp” e “Viewability” e ad agganciare la tabella dimensionale “Holding”. Oltre al “Codice Holding”, la “Link Grp-Viewability” contiene la chiave “KEY_LINK_TABLE_GRP_VW, costruita a partire dai campi “Codice Holding”, “Codice Raggruppamento Concessionari Nielsen”, “Codice Raggruppamento Mezzi Nielsen”, “Anno” e “Mese”. La chiave “KEY_LINK_TABLE_GRP_VW” viene utilizzata anche per connettere la link table “Link Investimenti-Mercato”. L’ultimo campo presente all’interno di questa tabella è “KEY_LINK_TABLE_BENCHMARK_CPG”, ossia una chiave necessaria per il collegamento con la link table “Link Cpg Benchmark”.
- Link Cpg Benchmark: questa è l’ultima delle tre tabelle di link e consente di connettere la fact table “Cpg 30” e le tabelle dimensionali “Concessionari”, “Calendario” e “Raggruppamento Mezzi” al resto della nuvola dati. Il campo chiave utilizzato per il collegamento con la link table “Link Grp-Viewability” è “KEY_LINK_TABLE_BENCHMARK_CPG” ed è ottenuto grazie alla concatenazione dei campi comuni a tutte le cinque tabelle dei fatti, ossia “Codice Raggruppamento Concessionari Nielsen”, “Codice Raggruppamento Mezzi Nielsen”, “Anno” e “Mese”.

Terminata la trattazione del modello dati relativo all'app "AQC", è possibile passare al modello dati creato a supporto dell'applicazione "Flow". Come si evince dalla figura sottostante, anche in questo caso si è scelto di strutturare il data model secondo lo "star schema", infatti è presente un'unica fact table centrale (Dati Nielsen), cui sono direttamente collegate otto tabelle dimensionali. La tabella dei fatti è stata realizzata caricando i dati presenti nella vista materializzata VDT_DATI_NIELSEN che, a differenza della VDT_INV_MERCATO, ospita le informazioni raccolte da Nielsen sugli investimenti dei clienti, dettagliandole a livello di marca, così da consentire un'analisi degli investimenti legata a singoli prodotti. Le misure contenute nella tabella "Dati Nielsen" sono tre, ossia "Netto Nielsen", "Netto Marketing Nielsen" e "Netto Accordo Quadro Nielsen" e sono declinate per "Codice Marca Nielsen", "Codice Holding", "Codice Raggruppamento Concessionari Nielsen", "Codice Raggruppamento Mezzi Nielsen", "Codice Rete Nielsen", "Codice Settore Nielsen", "Anno", "Mese" e "Settimana". Quasi tutte le tabelle dimensionali presenti sono una copia di quelle realizzate per il modello dati dell'app "AQC" e risulta perciò superfluo descriverle di nuovo. L'unica tabella che può essere considerata "nuova" è la "Marche Clienti Nielsen", la quale attinge dalla vista VDM_MARCHE_CLIENTI_NIELSEN e permette di stabilire una relazione tra le marche ed i clienti Nielsen che le producono. Tale tabella si aggancia alla fact table mediante il "Codice Marca Nielsen". Un'ulteriore differenza rispetto al data model dell'app "AQC" riguarda la tabella "Calendario", la quale ospita tutti i possibili valori temporali compresi tra la prima e l'ultima data individuate all'interno della fact table.

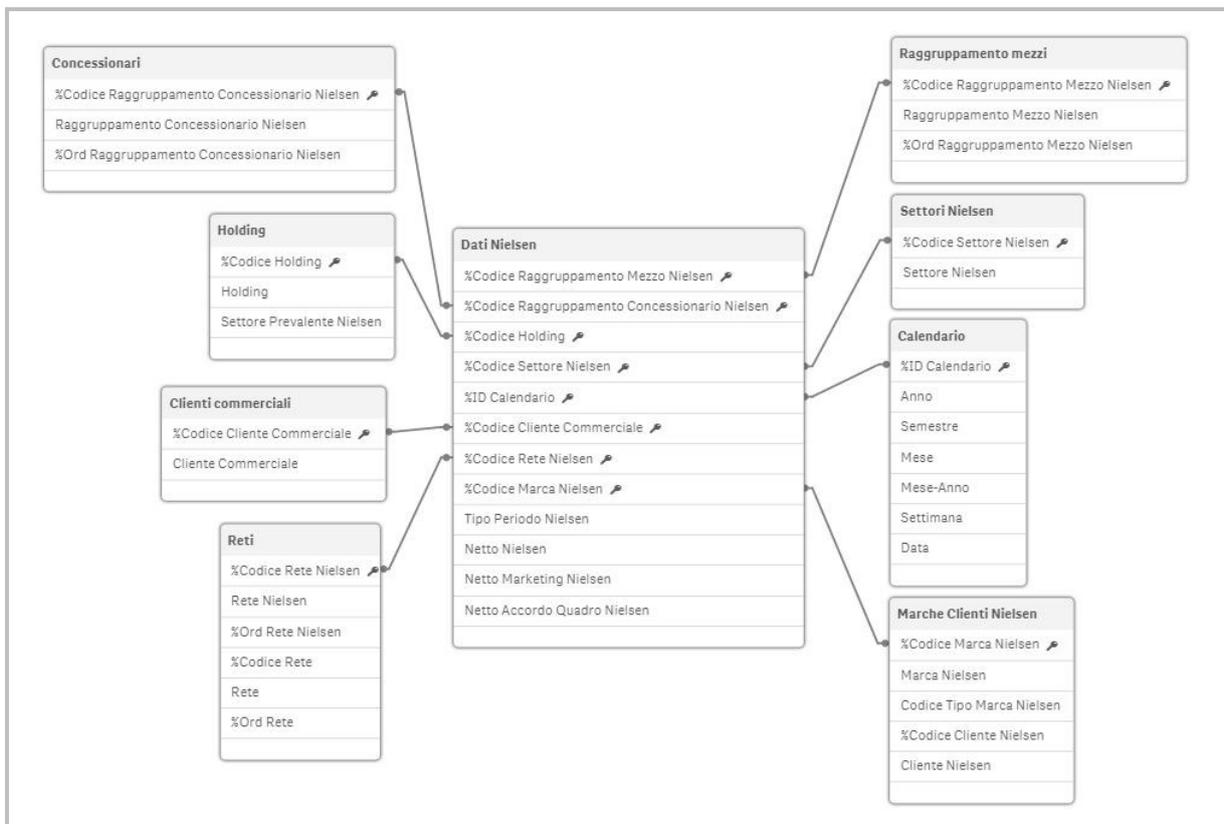


Figura 12 - Struttura del data model costruito per l'app "Flow"

3.7 Implementazione del front-end delle app

L'ultimo step da compiere sulla strada verso il completamento delle due app Qlik Sense è costituito dalla traduzione delle informazioni presenti all'interno dei data model in elementi grafici di impatto, che siano d'aiuto agli utilizzatori finali nell'esplorazione dei dati e nella conseguente identificazione di fenomeni e trend altresì invisibili. Questa fase, sebbene appaia meno consistente rispetto al lavoro svolto fino a questo punto, risulta importantissima in ambito BI, in quanto gli strumenti realizzati a livello grafico costituiscono l'unico punto di contatto tra i dati grezzi e gli utenti. L'insieme delle tecniche che consentono il passaggio di un dato dalla sua rappresentazione in forma testuale ad una rappresentazione realizzata mediante elementi visivi (puntini, barre, curve, etc.) prende il nome di "data visualization" o "dataviz" e si sviluppa a partire dall'idea secondo cui le immagini riescono ad essere elaborate dalla mente umana ad una velocità fino a 60000 volte maggiore rispetto al testo. Veicolare le informazioni attraverso le figure, dunque, può costituire un grosso vantaggio soprattutto in ambito aziendale, dove è sempre di meno il tempo a disposizione per condurre analisi e prendere decisioni.

Qlik Sense, come tool leader in ambito BI, offre una vastissima gamma di strumenti grafici interattivi, detti "visualizations", che possono essere inseriti all'interno delle dashboard, con l'obiettivo di presentare i dati in modo chiaro ed efficace. Le visualizations messe a disposizione da Qlik sono:

- Grafico a barre (bar chart): questo grafico è adatto per il confronto tra più valori. Su uno dei due assi viene posta la dimensione, sull'altro la misura, il cui valore varia a seconda del valore assunto dalla dimensione. Con il grafico a barre è anche possibile eseguire confronti più complessi di dati, utilizzando barre in pila o raggruppate, impiegando eventualmente due dimensioni ed una misura. Il raggruppamento e l'impilamento delle barre, molto spesso, semplificano la visualizzazione dei dati raggruppati.
- Grafico lineare (line chart): questo grafico è principalmente adatto quando si desidera visualizzare le tendenze e i movimenti nel corso del tempo, nelle situazioni in cui i valori sono distanziati in modo omogeneo, ad esempio distribuiti in mesi, trimestri o anni fiscali. La dimensione è collocata sempre nell'asse delle X, mentre le misure sono collocate sempre nell'asse delle Y. La serie di dati deve contenere almeno due punti dati per tracciare una linea. Una serie di dati con un singolo valore viene visualizzata come un punto.
- Grafico combinato: questo grafico è adatto per il confronto di due set di valori di misura generalmente difficili da confrontare a causa delle differenze di scala. Un esempio tipico è quando si desidera combinare le cifre di vendita in un grafico a barre con i valori del margine (in percentuale). In un grafico a barre standard le barre relative alle vendite vengono visualizzate normalmente, mentre i valori del margine risultano quasi invisibili a causa della grande differenza tra i valori numerici per le vendite e per il margine. Se si dispone di un'altra misura, ad esempio le vendite lorde, con valori che rientrano approssimativamente nella stessa scala dei valori di vendita, è possibile aggiungere la terza misura come barre impilando o raggruppando i nuovi valori di misura con i valori di vendita. Con le barre raggruppate è possibile confrontare facilmente due o più voci nello stesso gruppo categorico. Le barre in pila combinano barre di gruppi differenti una sull'altra e l'altezza totale della barra risultante rappresenta il risultato combinato. Grazie alla presenza di scale di misura diverse, una a sinistra e una a destra, il grafico combinato è ideale per quando si desidera presentare valori di misura generalmente difficili da combinare a causa della grande differenza negli intervalli di valori.

- Grafico a torta (pie chart): questo grafico visualizza la relazione esistente tra valori nonché la relazione di un singolo valore rispetto al totale. È possibile utilizzarlo quando si dispone di singole serie di dati che includono solo valori positivi. Nel grafico a torta le dimensioni formano i settori del valore della misura ed è sempre possibile includere una sola dimensione e una sola misura.
- Istogramma: questo grafico è adatto per visualizzare la distribuzione di dati numerici su un intervallo continuo o per un certo periodo di tempo. I dati sono divisi in contenitori e ogni barra dell'istogramma rappresenta la frequenza tabellare per ogni contenitore.
- Box plot: questo elemento grafico è adatto per il confronto degli intervalli e delle distribuzioni di gruppi di dati numerici ed è rappresentato da una scatola con baffi e una linea centrale. I baffi rappresentano i valori di riferimento alto e basso per escludere i valori anomali. È possibile definire i punti iniziale e finale della scatola e gli intervalli dei baffi con diversi set di impostazioni, oppure definire impostazioni personalizzate mediante espressioni.
- Grafico di distribuzione: questo grafico è adatto per il confronto degli intervalli e delle distribuzioni di gruppi di dati numerici. I dati sono tracciati come punti dei valori lungo un asse. È possibile scegliere di visualizzare solo i punti dei valori, per vedere la distribuzione dei valori, una casella di delimitazione, per vedere l'intervallo dei valori, oppure una combinazione dei due.
- Grafico a dispersione: Il grafico a dispersione presenta i valori di diverse misure su una dimensione come un insieme di punti. Nella maggior parte dei grafici la dimensione è collocata su uno degli assi, tuttavia, in un grafico a dispersione essa è rappresentata dai punti presenti nel grafico, mentre le misure sono collocate su uno degli assi. Quando si utilizza una terza misura (opzionale), il relativo valore viene riprodotto nella dimensione della bolla. Quando si analizzano serie di dati di grandi dimensioni e vengono visualizzati i dati compressi, la densità dei punti dati è riflessa dal colore.
- Grafico a cascata: questo grafico è adatto a illustrare come un valore iniziale viene influenzato da valori intermedi positivi e negativi. I valori iniziale e finale sono rappresentati da barre intere, mentre i valori intermedi da barre mobili. È anche possibile visualizzare i subtotali nel grafico. In un grafico a cascata è necessario utilizzare una misura per ogni barra del grafico. L'ordine delle misure definisce l'ordine delle barre nel grafico. Per ogni misura è necessario definire come influisce sul valore precedente.
- Mappa ad albero (tree map): questo grafico permette di visualizzare i dati gerarchici utilizzando rettangoli nidificati, ossia rettangoli di dimensioni inferiori all'interno di rettangoli più grandi.
- Mappa: questi oggetti possono essere utilizzati per una vasta gamma di scopi. Negli strumenti di BI vengono comunemente utilizzati per il tracciamento dei dati sulle vendite per regione o per negozio. È possibile creare una mappa utilizzando un livello punto (file Excel o KML) o un livello area (file KML).
- Misuratore (tachimetro): questo oggetto è progettato per mostrare un singolo valore di misura e visualizzare come interpretare tale valore.
- Tabella: questo oggetto mostra diversi campi contemporaneamente e, al suo interno, il contenuto di ogni riga è collegato logicamente. In genere, una tabella è costituita da una dimensione e da diverse misure, ed è possibile applicare selezioni solo nelle colonne delle dimensioni.

- Tabella pivot: questo oggetto mostra dimensioni e misure come righe e colonne all'interno di una tabella. In una tabella pivot è possibile analizzare i dati contemporaneamente in base a più misure e in più dimensioni. È possibile ridisporre le misure e le dimensioni per ottenere viste dei dati differenti. L'azione di spostamento di misure e dimensioni intercambiabilmente tra righe e colonne è denominata "pivoting".
- Testo e immagine: è un elemento complementare di altre visualizzazioni poiché offre opzioni per l'aggiunta di testo, immagini, collegamenti ipertestuali e misure. È possibile formattare e colorare il testo e allineare i paragrafi. L'immagine di sfondo presenta opzioni di ridimensionamento e posizionamento. È inoltre possibile impostare il comportamento reattivo per il testo e le immagini.
- KPI: mostra uno o due valori di misura e viene utilizzata per tenere traccia delle prestazioni. I KPI possono essere utilizzati per ottenere una panoramica dei valori delle prestazioni che sono fondamentali per un'organizzazione. È possibile utilizzare la combinazione di colori e i simboli per indicare la correlazione tra i numeri e i risultati previsti.
- Casella di filtro: consente di filtrare contemporaneamente i dati di più dimensioni. Se, ad esempio, si dispone di un grafico in cui sono riportate le vendite nel corso del tempo, è possibile utilizzare una casella di filtro per limitare i dati nel grafico affinché vengano visualizzate solo le vendite di un periodo di tempo selezionato, di una determinata categoria di prodotti e di una determinata regione.

Laddove nessuno degli oggetti appena elencati risultasse adatto alla rappresentazione di una determinata informazione, sarebbe possibile, attraverso le web API messe a disposizione da Qlik Sense, sviluppare componenti grafiche completamente nuove, chiamate “visualization extensions”, richiamabili all'interno delle dashboard. Le visualization extensions si basano su HTML, JavaScript e CSS e, una volta sviluppate, possono essere pubblicate su una piattaforma chiamata “Qlik Branch”, così da risultare disponibili per tutti gli utenti che volessero scaricarle ed utilizzarle.

Molti degli oggetti appena descritti sono interattivi e, di conseguenza, consentono agli utenti di effettuare selezioni sull'oggetto stesso. In Qlik le selezioni, come è stato ampiamente illustrato nel capitolo 2, provocano un ricalcolo istantaneo del modello dati sottostante all'applicazione e gli effetti di questo ricalcolo si ripercuotono su tutte le visualizations presenti all'interno dell'app. Esiste però una tecnica, detta “set analysis”, che consente, attraverso l'uso di un'apposita sintassi, di modificare l'insieme di dati cui una visualization attinge, rendendo di fatto l'oggetto indipendente dalla selezione. L'uso della set analysis risulta necessario, ad esempio, quando si vuole realizzare un confronto tra due valori di una stessa misura, uno variabile rispetto alla selezione e l'altro fisso. Per meglio comprendere il concetto di set analysis, è necessario fare un esempio:

$$sum(\{ \$ <Year=\{2009\}> \} Sales)$$

Tale stringa, se inserita come misura all'interno di una visualization, fa sì che il modello dati sottostante l'oggetto sia costituito dalle sole vendite dell'anno 2009 e, di conseguenza, qualunque sia la selezione dell'utente, questa potrà essere applicata solo ed esclusivamente al sottoinsieme di informazioni definito dalla set analysis. In questo caso particolare, ad esempio, le selezioni applicate dall'utente, relative all'anno, non avranno alcun effetto sull'oggetto considerato. La set analysis è costituita sempre da un'operazione di aggregazione, effettuata su una certa misura, e da una “set expression”. Relativamente all'esempio considerato, la set expression coincide con la seguente sottostringa:

$$\{ \$ <Year=\{2009\}> \}$$

Una set expression viene utilizzata all'interno di un'espressione di aggregazione per dare vita ad una stringa di set analysis e deve essere sempre compresa tra due parentesi graffe. All'interno di una set expression è possibile identificare le seguenti tre entità:

- **Identificatore:** gli identificatori definiscono la relazione tra la set expression e la misura sulla quale la set expression viene applicata. Nell'esempio $sum(\{\$<Year=\{2009\}>\}Sales)$, l'identificatore corrisponde al simbolo del dollaro '\$', e indica che la serie di record da valutare include tutti i record della selezione attuale. Un altro possibile identificatore è '1', il quale indica che la serie di record da valutare include tutti i record, indipendentemente dalla selezione. Esistono poi combinazioni che mettono assieme i due identificatori '\$' e '1', come ad esempio "\$1" o "\$_1" i quali servono ad indicare, rispettivamente, la selezione precedente e quella successiva.
- **Operatore:** gli operatori consentono di includere, escludere e intersecare parzialmente e interamente le serie di dati. Tutti gli operatori utilizzano i gruppi come operandi e restituiscono un gruppo come risultato. I possibili valori che un operatore può assumere sono: '+' (operatore di unione), '-' (operatore di esclusione), '*' (operatore di intersezione), '/' (operatore di differenza simmetrica o XOR).
- **Modificatore:** i modificatori consentono di apportare aggiunte o modifiche a una selezione. Tali modifiche possono essere scritte nella set expression. Un modificatore include uno o più nomi di campo, ciascuno seguito da una o più selezioni che possono essere effettuate nel campo. I modificatori iniziano e terminano con parentesi angolari, $\langle \rangle$.

Una volta compresa l'importanza dell'introduzione di elementi grafici, in luogo dei classici resoconti testuali, è necessario comprendere quale sia il modo migliore di presentare visivamente un certo insieme di informazioni. In linea generale, la rappresentazione deve essere accattivante e, allo stesso tempo, intuitiva ed è necessario esporre solamente le informazioni utili, scartando tutto ciò che risulta superfluo; avere troppe informazioni, infatti, potrebbe causare confusione nella mente degli utenti, i quali preferiscono che lo scopo di una visualizzazione risulti chiaro fin dall'inizio. Nel caso specifico dell'app "AQC", si è deciso, in linea con quanto suggerito dal committente, di suddividere le informazioni da esporre in base all'ambito di interesse, realizzando di fatto una dashboard (foglio) diversa per ciascun "argomento". Complessivamente sono stati realizzati sei fogli, aventi le finalità indicate di seguito:

- Foglio "Selezione cliente": consente all'utente di selezionare un cliente o una holding rispetto cui condurre l'analisi.
- Foglio "Scheda informativa": fornisce all'utente la possibilità di conoscere il comportamento del cliente in modo generico.
- Foglio "Investimenti Rai Pubblicità": consente all'utente di approfondire la conoscenza del cliente attraverso l'analisi degli investimenti effettuati da quest'ultimo in ambito Rai Pubblicità.
- Foglio "Analisi mercato": consente all'utente di approfondire la conoscenza del cliente attraverso l'analisi degli investimenti effettuati da quest'ultimo relativamente ai principali competitors.
- Foglio "Focus TV/Radio": fornisce all'utente la possibilità di approfondire la conoscenza del cliente, soffermandosi sugli investimenti che costui effettua relativamente ai soli mezzi TV e radio.

- Foglio “Focus web”: fornisce all’utente la possibilità di approfondire la conoscenza del cliente, soffermandosi sugli investimenti che costui effettua relativamente al mondo internet.

Per quello che concerne l’app “Flow”, invece, questa risulta costituita da un unico foglio, finalizzato, conformemente allo scopo dell’applicazione, all’analisi degli investimenti del cliente rispetto ai singoli prodotti.

Nella disporre le visualizations all’interno dei fogli sono state seguite alcune best practices, tra cui il posizionamento dei filtri nella parte alta dello schermo (posizione prioritaria verso la quale gli utenti sono più propensi a rivolgere lo sguardo), l’uso di dimensioni diverse per comunicare all’utente la differenza tra informazioni più importanti (quelle più grandi) e meno importanti (quelle più piccole) e l’uso dei colori verde e rosso per indicare rispettivamente valori positivi e valori negativi.

In aggiunta a ciò, è stato necessario considerare, in fase di disposizione degli elementi, il possibile scenario che si sarebbe delineato nel caso di utilizzo delle app su dispositivi mobili, quali smartphone e tablet. In questo scenario, infatti, Qlik dispone gli oggetti, in modo automatico, uno sotto l’altro, partendo dagli oggetti che nella versione desktop occupano la posizione in alto a sinistra, fino ad arrivare agli oggetti che si trovano in basso a destra. Per questo motivo, dunque, si è cercato di mantenere vicine, per quanto possibile, le visualizations relative ad una stessa informazione, così che queste potessero risultare adiacenti anche nel formato mobile.

Prima di procedere alla realizzazione di ciascuno dei fogli indicati sopra è stato necessario creare quella che in Qlik viene definita “master library”, ossia un insieme di misure e dimensioni che risultano di particolare interesse per tutte le dashboard e che, essendo definite a priori, possono essere facilmente richiamate nella fase di creazione delle visualizations. Le misure e le dimensioni inserite all’interno della master library prendono, rispettivamente, i nomi di “master measures” e “master dimensions” ed il loro principale vantaggio risiede nel fatto che, qualora fosse necessario apportare modifiche ad una misura o ad una dimensione, basterebbe andare ad operare solo sullo specifico master item, senza coinvolgere le componenti grafiche.

A partire dalla versione più recente, inoltre, Qlik Sense fornisce agli sviluppatori la possibilità di definire, per le master dimension che comprendono pochi valori distinti, il colore da assegnare a ciascun valore quando la dimensione viene inserita all’interno di un grafico. Nell’ambito delle app “AQC” e “Flow”, si è scelto di servirsi di questa funzionalità per definire a priori i colori da assegnare ai diversi valori delle dimensioni “Mezzo” e “Concessionario”, prendendo in considerazione l’idea di creare delle palette di colori fisse, da utilizzare anche durante lo sviluppo delle future applicazioni. Sempre a proposito dei colori, poi, si è deciso di evitare l’uso di troppe tinte diverse all’interno della stessa app, in quanto ciò potrebbe causare una maggiore confusione nella mente dell’utente.

Una volta completata la creazione della master library, è stato utile creare le variabili, ossia entità denominate, contenenti valori di dati, che possono essere impiegate internamente alle definizioni delle misure e delle dimensioni, al fine di semplificarne la scrittura.

Nel seguito di questo paragrafo saranno analizzati da un punto di vista sia grafico che funzionale le dashboard attraverso cui sono articolate le app “AQC” e “Flow”, rivolgendo particolare attenzione, laddove fosse necessario, alle singole visualizations utilizzate nella composizione dei fogli.

3.7.1 Foglio 1 – “Selezione cliente”

Come è possibile evincere dall’immagine, l’elemento principale di questo foglio è un grafico a barre, finalizzato alla rappresentazione dei clienti, ordinati in base alla somma degli investimenti da loro effettuati negli ultimi cinque anni (dal 2014 al 2018). Il grafico utilizza la tecnica delle barre sovrapposte per combinare la dimensione temporale “Anno” con quella “Holding”, infatti, nonostante sia presente un’unica barra per ogni cliente, questa è colorata in modo diverso in base all’anno cui gli investimenti si riferiscono. Sopra il bar chart, poi, è stata inserita una casella di testo dinamica contenente la ragione sociale della holding selezionata o, in assenza di una selezione, la stringa “selezionare una sola holding/cliente”. Nel caso in cui l’utente decidesse di selezionare più di una holding, dal momento che lo scopo di questa applicazione è quello di condurre l’analisi rispetto ad un solo cliente alla volta, nella casella di testo sarebbe visualizzata nuovamente la stringa “selezionare una sola holding/cliente”.

Un’altra componente importantissima è costituita dall’insieme di filtri posto nella parte alta del foglio; questo oggetto costituirà un leitmotiv per questa applicazione e sarà riproposto anche nelle dashboard successive, in quanto si vuole che l’utente conduca la propria analisi proprio attraverso l’uso dei filtri. Risulta necessario approfondire la finalità dei filtri “Settore Prevalente” e “Accordo Quadro Si/No”: il primo può tornare utile qualora si volessero visualizzare solamente le holding i cui investimenti ricadono, per la maggior parte, in un dato ambito; il secondo, invece, permette di escludere dall’analisi le holding non idonee alla stipula di un accordo quadro.

Ultimo elemento di rilievo all’interno di questo foglio è il gruppo di “bottoni” posto nella parte in basso. Tali oggetti non sono altro che visualization extensions inserite per consentire all’utente di “saltare” ad uno qualsiasi degli altri fogli che compongono l’app “AQC”. L’ultimo bottone, il quale presenta la label “Flow”, è diverso dagli altri, poiché permette di “saltare” al di fuori dei confini dell’app corrente, catapultando l’utente all’interno dell’applicazione “Flow”. La visualization extension utilizzata è la “Sheet Navigation & Actions” e, oltre a permettere il passaggio da un foglio ad un altro, all’interno del medesimo applicativo, consente anche collegamenti intra app che sfruttano le URL. Ogni applicazione è identificata in modo univoco da una URL ed è possibile, grazie alle “Integration API” messe a disposizione da Qlik, concatenare la base URL con le eventuali selezioni effettuate dall’utente, così che, nel passaggio da un’app all’altra tali selezioni non vadano perse.

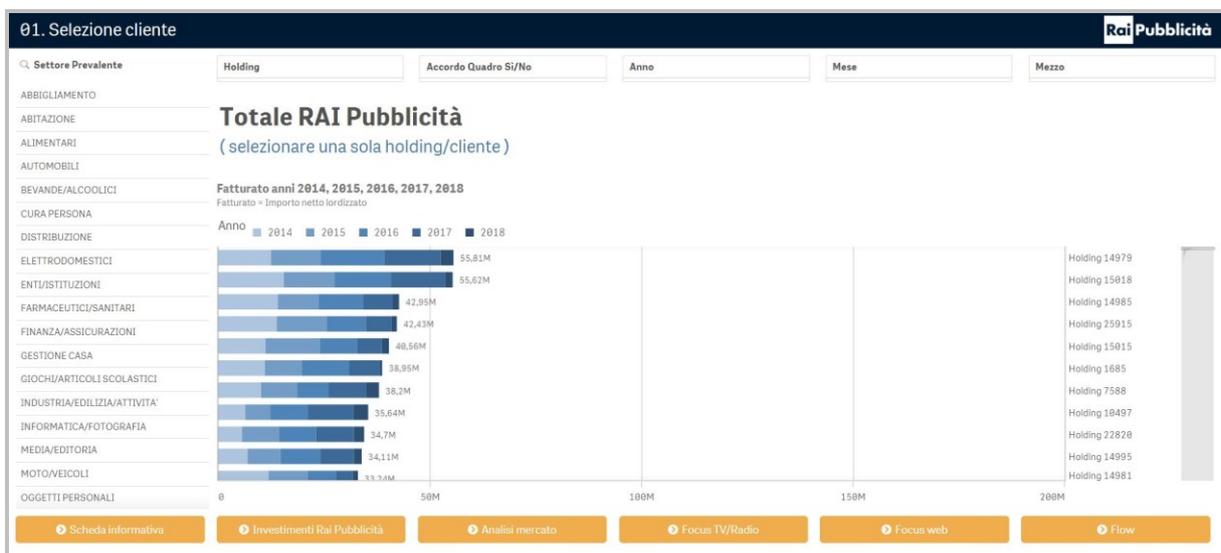


Figura 13 - Foglio "Selezione Cliente"

3.7.2 Foglio 2 – “Scheda informativa”

Una volta selezionata la holding di interesse, l’utente può passare al secondo foglio che, come già detto, fornisce una serie di informazioni generiche. Innanzitutto, in alto a sinistra, è stata riproposta la casella di testo contenente la ragione sociale del cliente, il settore merceologico di riferimento e la fonte dati cui la dashboard attinge; è importante notare che il testo presenta il medesimo stile utilizzato in precedenza, così da creare una certa continuità con il primo foglio. Sul lato opposto, sono stati inseriti due KPI utili ad indicare la posizione occupata dal cliente selezionato all’interno del ranking complessivo (posizione del cliente rispetto a tutti gli altri clienti di Rai Pubblicità) e quella calcolata rispetto agli altri clienti con cui condivide il settore prevalente. La classifica è stilata in base agli investimenti che i clienti hanno effettuato in Rai e fa fede ai dati raccolti dalla Nielsen.

A livello sottostante è possibile trovare, invece, tre visualizations, ossia un bar chart che, come suggerisce il titolo, espone gli investimenti del cliente divisi per anno e due grafici a torta, i quali rappresentano rispettivamente come sono distribuiti gli investimenti del cliente rispetto ai mezzi e ai settori. Sia il bar chart che il primo dei due pie chart (quello a sinistra) sono di tipo “drill-down”, ossia contengono, oltre alla dimensione principale, una dimensione aggiuntiva, che permette all’utente di condurre l’analisi ad un livello di dettaglio maggiore.

Dal momento che risulterebbe inutile comparare l’anno corrente con gli anni precedenti, in quanto per l’anno corrente mancano le informazioni sui mesi che ancora non sono trascorsi, si è pensato di inserire all’interno del foglio il filtro “Year To Month”, il quale permette all’utente di escludere dal confronto gli investimenti relativi a quei mesi che non sono comuni a tutti e cinque gli anni.

Ancora più in basso è presente una tree map che, nel caso in cui l’analisi fosse condotta rispetto ad una holding, permetterebbe di identificare i diversi clienti che costituiscono la holding e di distinguerli in base agli investimenti (più il cliente ha investito e più risulta grande il quadrato che lo rappresenta). Nell’immagine presa come esempio la holding selezionata è formata da un unico cliente ed è per questo motivo che viene visualizzato un solo riquadro. Per concludere, sul fondo del foglio, sono presenti tre tabelle, contenenti rispettivamente le informazioni anagrafiche relative alle agenzie, ai venditori di agenzia e ai venditori della clientela che si occupano del cliente selezionato.

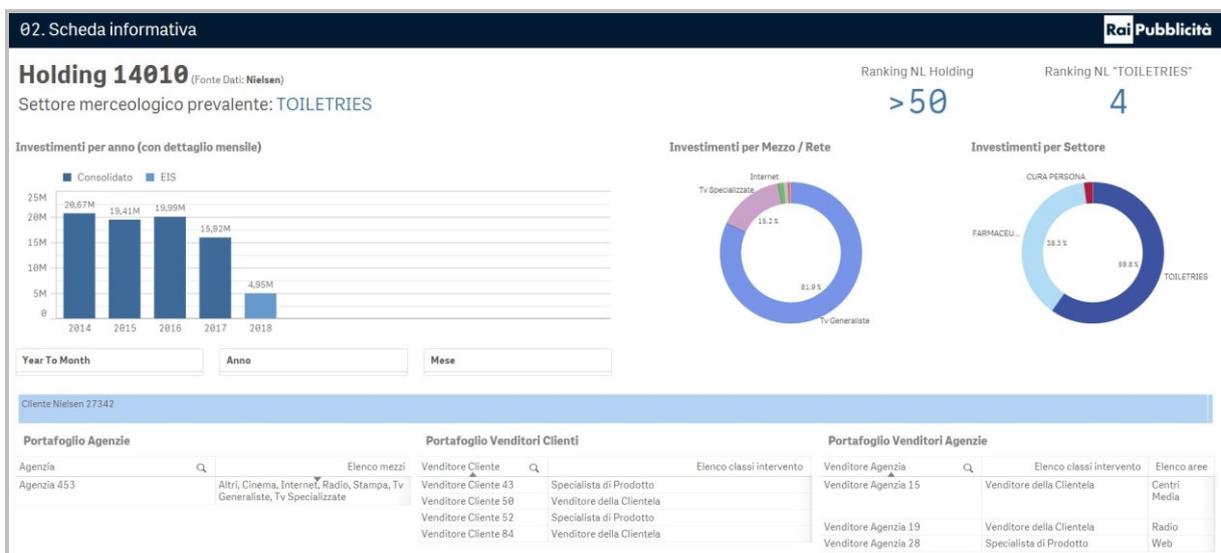


Figura 14 - Foglio "Scheda informativa"

3.7.3 Foglio 3 – “Investimenti Rai Pubblicità”

Questo foglio, a differenza del precedente, attinge alle informazioni raccolte da Rai Pubblicità (la fonte, come per il foglio 2, è specificata a destra della ragione sociale del cliente) e si sofferma sugli investimenti effettuati dal cliente per acquistare spazi pubblicitari in Rai.

Partendo dall’alto, il primo elemento incontrato è l’insieme delle caselle di filtro, contenente i filtri “Holding”, “Cliente”, “Tipo Vendita”, “Evento”, “Mezzo”, “Rete”. “Tipo Vendita” ed “Evento” sono legati tra loro, in quanto il primo permette di scegliere se visualizzare solo le vendite standard o solo quelle relative agli eventi, mentre il secondo consente di selezionare, tra le vendite evento, solamente quelle legate ad una manifestazione specifica.

Oltre ai filtri, nel foglio compaiono nuovamente sia la text box contenente il nome del cliente selezionato, sia i KPI relativi al ranking. A differenza dei KPI visti in precedenza, però, questi espongono la posizione del cliente calcolata rispetto ai dati sugli investimenti raccolti da Rai Pubblicità e non rispetto alle informazioni fornite da Nielsen, ragion per cui il valore visualizzato in questa dashboard potrebbe non coincidere con quello contenuto nel foglio precedente.

Immediatamente sotto, poi, è possibile notare un bar chart analogo a quello già inserito all’interno della “scheda informativa”. Anche in questo caso il grafico risulta uguale al precedente solo nella forma, mentre differisce nei valori rappresentati. Il bar chart fornisce una rappresentazione degli investimenti che risulta molto intuitiva, ma potrebbe risultare scomoda agli utenti Rai, i quali sono stati abituati ad avere sempre un supporto di tipo tabellare. A tal proposito, è stato deciso di aggiungere al foglio una tabella pivot che, oltre a riportare in formato numerico le informazioni già contenute all’interno del grafico, potesse contenere una serie di misure di interesse aggiuntive, quali la percentuale di sconto applicata al cliente, la percentuale di sconto totale, il valore delle sanatorie concesse e gli omaggi contrattuali elargiti.

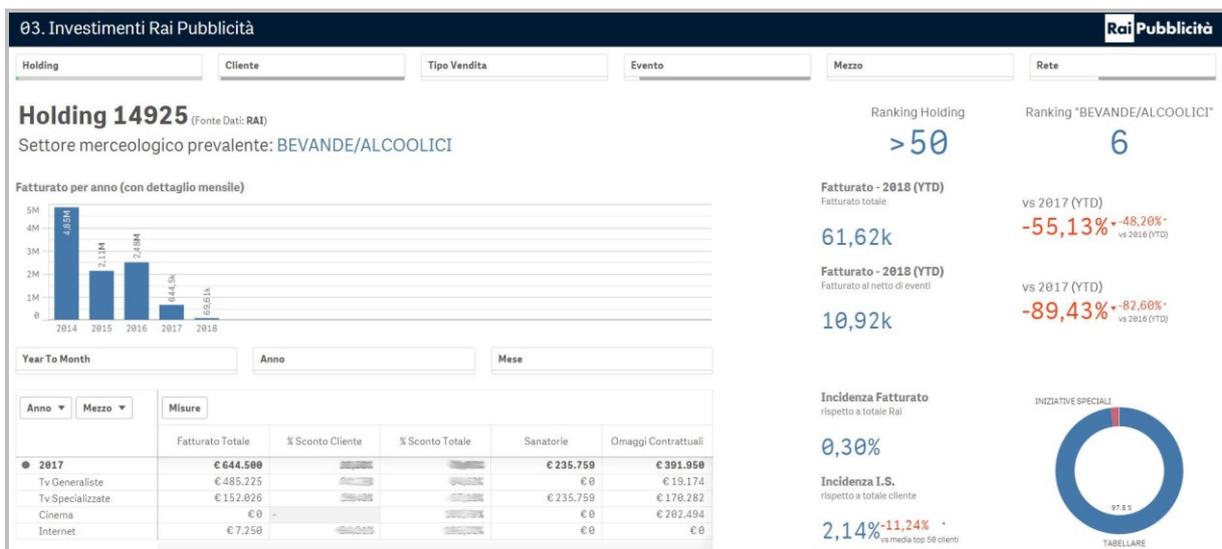


Figura 15 - Foglio "Investimenti Rai Pubblicità"

Sul lato destro del foglio è possibile notare la presenza di una serie di KPI: i quattro posizionati più in alto sono semanticamente legati tra loro e indicano il valore totale degli investimenti effettuati dal cliente, calcolato sia tenendo conto degli eventi (prima riga), sia escludendo dal calcolo gli investimenti legati alle manifestazioni (seconda riga). Per ciascuna delle due righe, poi, è esposto un valore percentuale che rende immediatamente comprensibile all'utente la differenza tra i totali calcolati rispetto all'anno corrente e quelli calcolati rispetto ai due anni precedenti. I KPI posizionati più in basso, invece, sono legati al pie chart e indicano quanto gli investimenti del cliente incidano sul fatturato totale Rai e che percentuale degli investimenti del cliente sia dedicata alle "iniziative speciali".

3.7.4 Foglio 4 – "Analisi Mercato"

Sorvolando sull'insieme di caselle di filtro e sulla text box contenente la ragione sociale del cliente selezionato, gli elementi di interesse di questa dashboard sono la tabella pivot, i due grafici a torta ed i nove KPI posizionati in basso.

La pivot table declina gli investimenti del cliente rispetto ai concessionari (righe) e all'anno (colonne). I concessionari considerati sono i principali competitors di Rai Pubblicità, ossia Mediaset ("PUBLITALIA*SEGRATE MI"), Sky ("SKY PUBBLICITA'MILANO"), La7 ("CAIRO COMMUNICATION MILANO") e Discovery Media ("DISCOVERY MEDIA MILANO").

A destra della tabella pivot sono presenti due pie chart: il primo contiene esattamente le stesse informazioni esposte dalla tabella, mentre il secondo è una visualizzazione di tipo "drill down" e permette all'utente di conoscere come gli investimenti del cliente sono distribuiti rispetto ai mezzi (TV, Radio, Internet, etc.) e, una volta selezionato un mezzo, gli consente anche di comprenderne la distribuzione rispetto alle reti relative a quel mezzo.

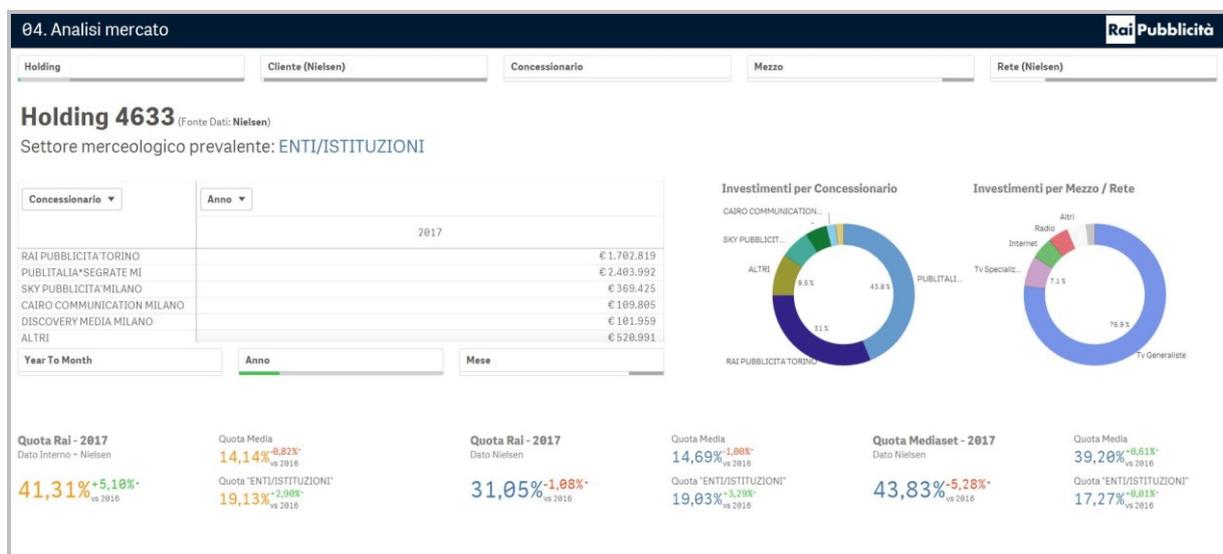


Figura 16 - Foglio "Analisi mercato"

I KPI posizionati in basso, invece, sono dedicati ad un confronto tra Rai Pubblicità e Mediaset e possono essere suddivisi in tre gruppi:

- Il primo gruppo espone, in percentuale, la quota di investimenti che il cliente ha deciso di investire in Rai, la quota media di investimenti calcolata facendo riferimento a tutti i clienti di Rai Pubblicità e la quota media di investimenti calcolata facendo riferimento ai soli clienti che hanno il settore merceologico di riferimento in comune con il cliente

selezionato. Ciascuno di questi tre valori è poi accompagnato da un delta percentuale che serve ad indicare la differenza rispetto allo stesso valore, calcolato per l'anno precedente. Tutte e tre le quote esposte sono ricavate utilizzando come fonte le informazioni raccolte internamente da Rai.

- Il secondo gruppo espone, sempre in percentuale, le medesime quote indicate per il primo gruppo, ma anziché attingere al dato interno, si serve delle informazioni fornite da Nielsen. Il dato reale, quindi, è quello presentato dal primo gruppo, mentre questo costituisce soltanto una stima.
- Il terzo gruppo espone invece la quota cliente, la quota media e la quota media calcolata rispetto al settore relative al principale competitor di Rai Pubblicità, ossia Mediaset. Anche in questo caso la fonte utilizzata è la Nielsen e, dunque, anche queste informazioni costituiscono una stima. Solitamente vengono messe a confronto le quote calcolate a partire dai dati Nielsen, ma avere in questo stesso foglio anche i dati reali può dare all'utente un'idea dell'errore legato alla stima.

3.7.5 Foglio 5 – “Focus TV/Radio”

All'interno del presente foglio sono presenti le informazioni su GRP, CPG e posizioni pregiate, ossia tutte informazioni che la Nielsen non è in grado di fornire e che, di conseguenza, come è possibile apprendere anche dalla dicitura alla destra della ragione sociale del cliente, vengono prelevate dal datawarehouse aziendale. Al solito, nella parte alta della pagina, è possibile trovare il set di filtri messi a disposizione dell'utente e la text box contenente nome del cliente e settore merceologico di riferimento.

La parte sottostante, invece, può essere suddivisa in due righe: sulla prima riga si trovano le informazioni che riguardano sia la TV che la radio (posizioni pregiate e redditività), mentre sulla seconda riga sono esposti dati relativi solo al mezzo TV (GRP e CPG). Partendo da sinistra, le visualization inserite sono:

- Tabella “Posizioni Pregiate TV/Radio”: è una tabella che, per ogni mezzo, indica il numero di secondi, la percentuale di spot trasmessi in una posizione pregiata, ossia nei primi o negli ultimi secondi di un break pubblicitario, e la percentuale di secondi di messa in onda rispetto al totale dei secondi.
- Bar chart “Redditività netta per anno”: è un bar chart di tipo drill-down che declina la redditività (rapporto tra investimento e secondi) rispetto alla dimensione “Anno”. Il fatto che il grafico sia di tipo drill-down vuol dire che, una volta selezionato un valore di anno, la declinazione della misura passerà ad un livello di dettaglio maggiore, in questo caso mensile.
- Bar chart “Redditività lorda per anno”: è un bar chart analogo a quello appena descritto, l'unica differenza risiede nel fatto che, mentre nel primo grafico era utilizzato l'importo netto lordizzato per il calcolo del valore di redditività, ora viene preso in considerazione l'importo lordo.
- Tabella “Valori GRP e CPG per target”: questa tabella serve ad indicare i valori di GRP e CPG legati a ciascun target. Affinché il contenuto degli ultimi due grafici a barre possa essere visualizzato, l'utente dovrà selezionare, direttamente sulla tabella, uno dei target disponibili.
- Bar chart “GRP per anno – Dato TV”: è un grafico a barre, di tipo drill-down, utile a rappresentare il valore di GRP declinato per anno o per mese.

Bar chart “CPG 30” per anno – Dato TV”: è un grafico combinato, di tipo drill-down, che permette di visualizzare i valori di CPG 30” declinati per anno o per mese, e di confrontarli con i valori di CPG 30” di benchmark, calcolati sia per Rai che per Mediaset. I valori di benchmark, come è possibile evincere dall’immagine di esempio, sono rappresentati mediante due linee e sono visibili soltanto se il target selezionato dall’utente è un target di tipo standard.

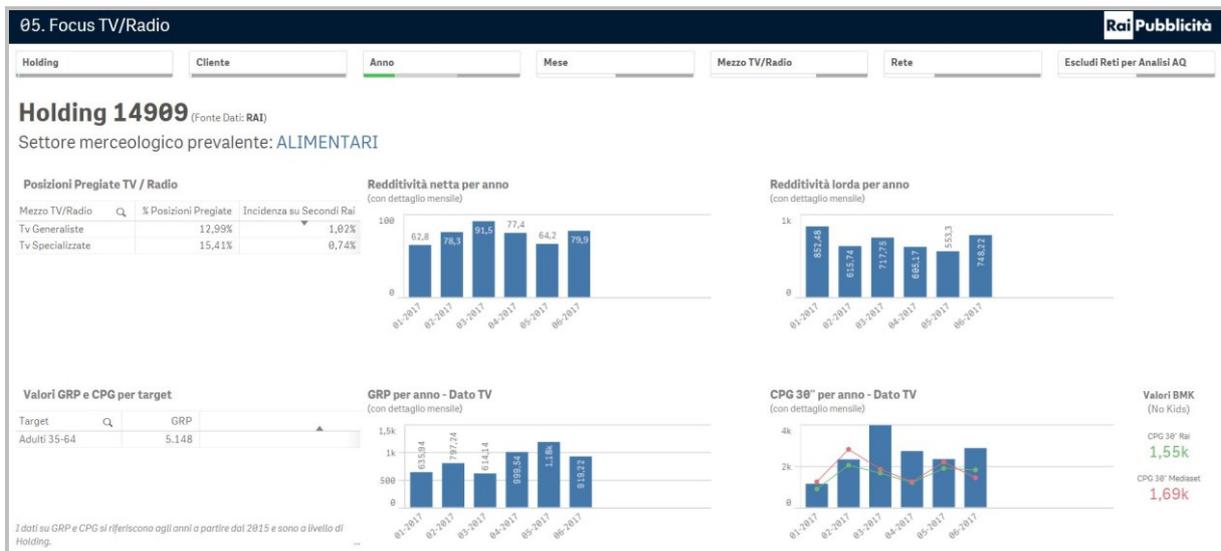
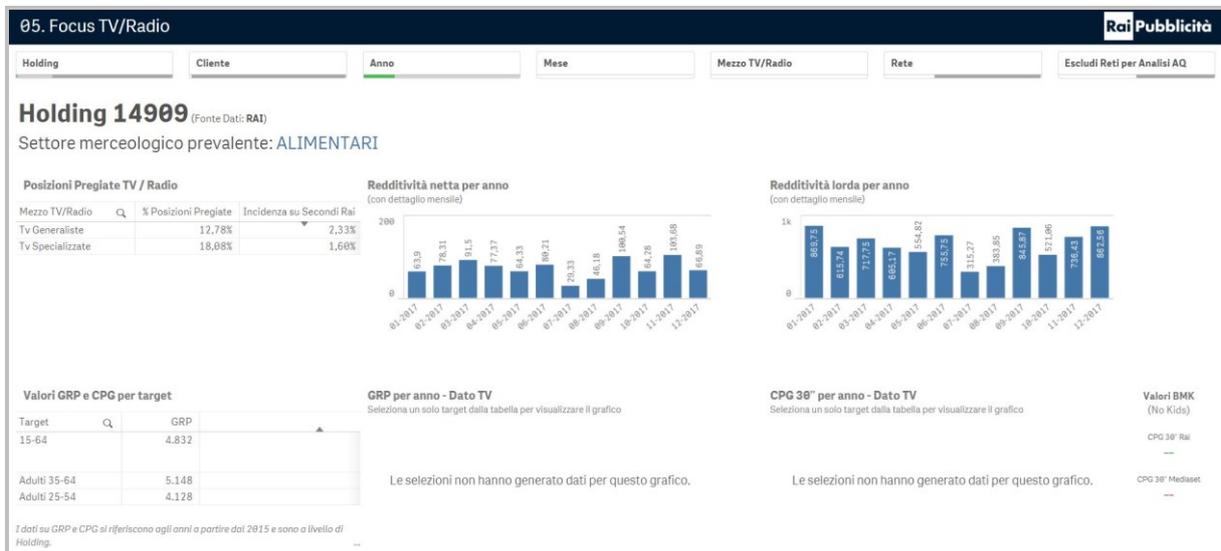


Figura 17 - Foglio "Focus TV/Radio", nessun target selezionato (in alto), target “Adulti 25-54” selezionato (in basso)

3.7.6 Foglio 6 – “Focus web”

Come nei precedenti fogli, anche in questo foglio compaiono sia i filtri che la text box contenente la ragione sociale del cliente ed il relativo settore merceologico prevalente. In aggiunta a queste visualizations, troviamo due line chart, utilizzati per rappresentare, rispettivamente, l’andamento del valore di CPM Medio e quello di CPM medio effettivo.

Ciascuno dei due grafici lineari rappresenta sia il valore di CPM relativo al cliente selezionato, sia quello calcolato rispetto alla media di tutti i clienti Rai, differenziando i due valori mediante l'uso di colori differenti.

Alla destra di ciascun line chart è stato aggiunto un KPI contenente il valore di scarto, espresso in percentuale, tra il CPM Medio calcolato per il cliente selezionato

Sotto i due line chart si trovano, poi, un grafico a barre che, grazie all'uso delle misure alternative, permette all'utente di scegliere se visualizzare i valori di impression vendute o quelli di impression erogate, declinati per anno, e una serie di KPI e text box contenenti le informazioni, raccolte da ComScore, relative al cliente e alla viewability. Nel dettaglio, risultano esposte la ragione sociale del cliente ed il settore merceologico di riferimento, secondo l'anagrafica ComScore, dopodiché viene messo in evidenza il valore di viewability relativo al cliente selezionato e quello relativo al settore merceologico cui il cliente appartiene, sottolineando la differenza percentuale che intercorre tra l'uno e l'altro.

Una nota posta in basso a destra indica che i valori di viewability sono disponibili solo per il primo semestre del 2017, questo perché, come già sostenuto in precedenza, tali valori sono estratti da un file Excel estemporaneo, che dovrà poi essere sostituito da una vista sul datawarehouse.

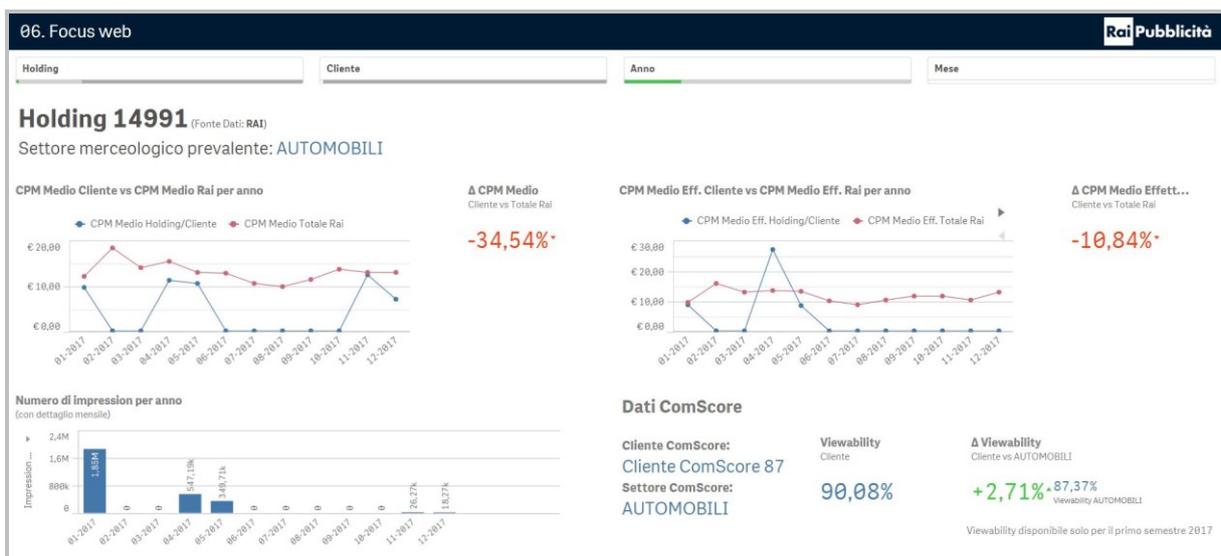


Figura 18 - Foglio "Focus web"

3.7.6 "Flow"

Come anticipato, l'app dedicata al flow consta di un unico foglio, denominato per l'appunto "Flow", il quale contiene, oltre al classico set di filtri e alla text box con il nome del cliente ed il settore merceologico di riferimento, le visualizations descritte di seguito.

- Una "tabella delle marche", contenente, ordinati per investimento, tutti i prodotti su cui i clienti abbiano deciso di investire negli ultimi cinque anni. Per ciascun prodotto è indicato, secondo l'anagrafica Nielsen, anche il nome del cliente.
- Un grafico a torta denominato "Investimenti per concessionario", indicante il modo in cui sono distribuiti gli investimenti del cliente sui diversi concessionari disponibili.

- Un grafico a torta, di tipo drill-down, denominato “Investimenti per Mezzo/Rete”, indicante il modo in cui sono distribuiti gli investimenti del cliente sui diversi mezzi e sulle diverse reti disponibili.
- Un line chart all’interno del quale ciascuna linea rappresenta un concessionario e che serve a declinare, settimana dopo settimana, il modo in cui il cliente selezionato ha deciso di investire. Il dettaglio settimanale, assieme alle informazioni sul singolo prodotto, è la motivazione principale che ha portato alla separazione dell’app “Flow” dall’app “AQC”.

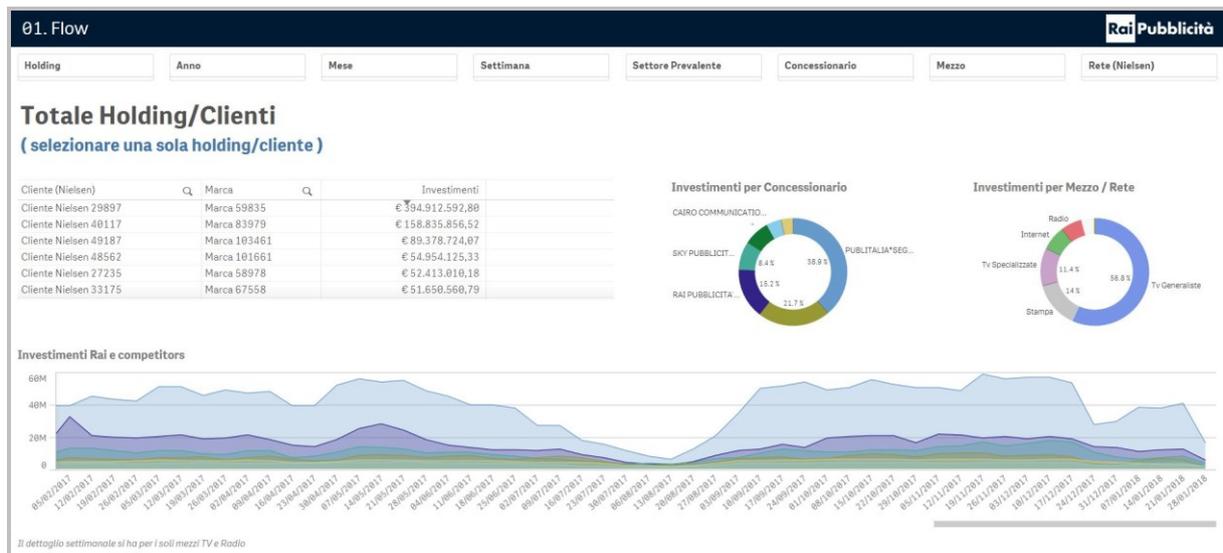


Figura 19 - Applicazione "Flow"

4. Realizzazione di un Chat Bot Telegram

Questo capitolo, con il quale si conclude l'elaborato di tesi, serve ad illustrare in maniera dettagliata il procedimento che ha portato alla realizzazione di un chat bot aziendale, che potesse permettere agli utenti di accedere facilmente alle informazioni contenute nelle due app Qlik Sense sviluppate in precedenza. Dopo aver indicato il funzionamento generico del bot e quelle che sono state le scelte progettuali adottate, si passerà ad un'analisi del software da un punto di vista implementativo, descrivendo le classi ed i metodi utilizzati. Naturalmente, il codice sviluppato è di carattere generico e necessita di un lavoro di configurazione, ogniqualvolta si presenti la necessità di supportare una nuova applicazione. A tal proposito, nella parte finale del capitolo, sarà posta l'attenzione sul processo di customizzazione che ha permesso di interfacciare il chat bot con l'app "Flow".

4.1 Scelte progettuali

Per prima cosa è stato necessario scegliere il linguaggio da utilizzare per sviluppare il back-end relativo al bot. La scelta è ricaduta sul C# e sul framework .NET, i quali consentono, grazie alle librerie messe a disposizione da Qlik Sense e da Dialogflow, di realizzare un software unico, in grado di gestire sia la comunicazione con le applicazioni Qlik, sia le operazioni di Natural Language Processing. Grazie alle api esposte da Telegram, inoltre, è stato possibile includere il codice relativo allo scambio di messaggi tra il server ed i client Telegram.

Per l'attività di sviluppo si è scelto di utilizzare l'editor Visual Studio (edizione 2017), il quale, grazie alla presenza di un tool per la gestione dei pacchetti (NuGet), ha semplificato il processo di integrazione delle api. NuGet, attraverso l'uso di un'apposita interfaccia, ha consentito la ricerca delle librerie necessarie e la conseguente aggiunta di queste al progetto. Nello specifico le librerie aggiunte, escludendo quelle utili alle operazioni di sistema, sono indicate di seguito:

- `Api.AiSDK`: tale libreria espone un insieme di metodi utili a supportare l'interazione tra il back-end e gli agent NLP configurati in Dialogflow.
- `QlikSense.NetSDK`: questa libreria mette a disposizione i metodi necessari ad interagire sia con la versione desktop che con la versione server di Qlik Sense e consente, oltre ad effettuare operazioni grafiche sulle app (creazione di nuovi fogli e visualizations, creazione di master items, etc.), di estrarre le informazioni direttamente dal modello dati sottostante una certa applicazione.
- `Telegram.Bot`: è una libreria contenente tutte le funzioni che permettono la connessione con i server Telegram. Come detto nel capitolo 2, i server Telegram fungono da intermediari tra i client ed il bot, ed è quindi essenziale non solo instaurare una connessione tra il bot ed il server, ma anche implementare operazioni che permettano al back-end di ricevere i messaggi inviati al server dagli utenti.

Data la possibilità, fornita dalle api, di creare un collegamento sia con la versione desktop di Qlik Sense che con quella server, si è deciso, per ragioni burocratiche, interne all'azienda, di permettere al bot di accedere soltanto alla versione desktop, contenente un sottoinsieme delle app disponibili sul server aziendale.

4.2 Architettura e implementazione del Chat Bot

Dato l'elevato numero di funzionalità da implementare, si è deciso di suddividere il back-end in moduli, facendo in modo che ciascun modulo sia dedicato, in modo esclusivo, all'esecuzione di un certo tipo di operazioni e che sia in grado di lavorare in maniera indipendente, rispetto al resto del programma. Un'architettura di questo tipo punta alla flessibilità, in quanto cerca di astrarre dalle specifiche tecnologie utilizzate. Qualora si decidesse, ad esempio, di sostituire una delle librerie utilizzate, infatti, basterebbe sostituire soltanto il modulo ad essa relativo, senza la necessità di dover ripensare da capo l'intero codice. I moduli identificati in fase di progettazione sono cinque, ossia "Modulo Bot", "Modulo Conversation", "Modulo NLP", "Modulo Qlik Sense" e "Modulo Log".

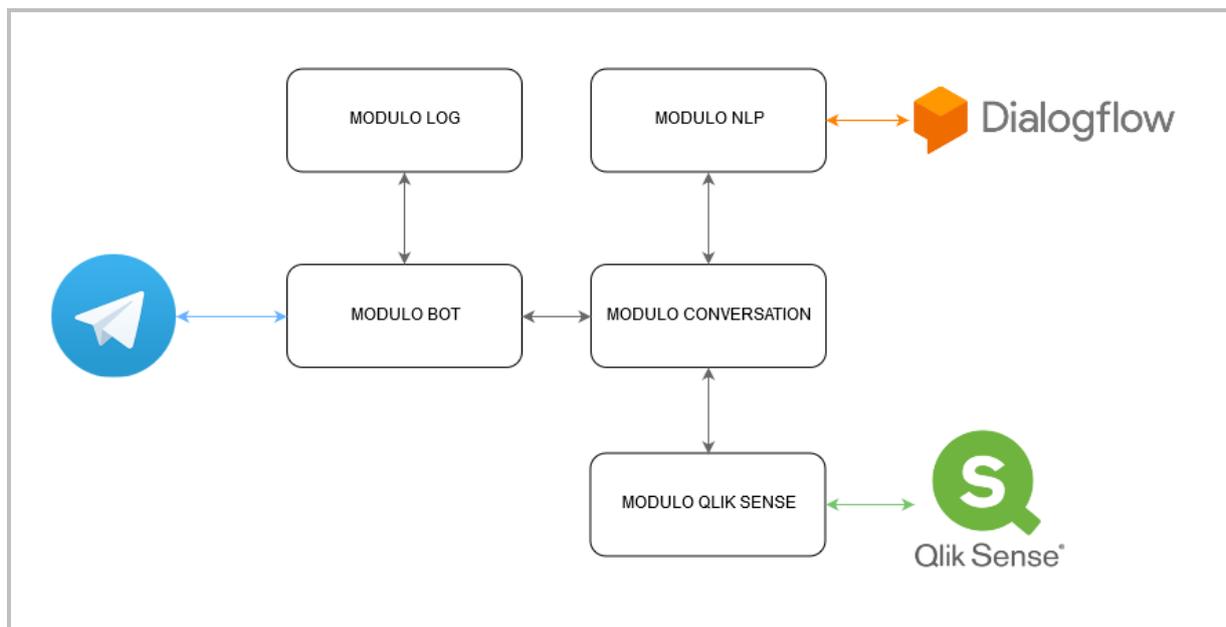


Figura 20 - Architettura del Chat Bot

4.2.1 Modulo Bot

Questo blocco costituisce il cuore del back-end, infatti, ha lo scopo di mettere in comunicazione tutti gli altri moduli. Per prima cosa, al suo interno, grazie alle api messe a disposizione dalla libreria TelegramBot, viene creata una nuova istanza della classe *TelegramBotClient*; questo oggetto permette al back-end di stabilire una connessione con il bot registrato sui server Telegram e di scambiare messaggi con i client. Fatto ciò, vengono definite due callback, ossia la *BotOnMessageReceived* e la *BotOnCallbackQueryReceived*, che vengono attivate, rispettivamente, quando viene ricevuto un messaggio e quando viene ricevuta la query corrispondente ad un "callback button". Prima di avviare la routine di ricezione dei messaggi, vengono creati anche altri tre oggetti: un'istanza della classe *QsApp*, utile a connettersi a Qlik Sense e ad interagire con le app, un'istanza della classe *TelegramConversation*, la quale sarà utilizzata per gestire la parte relativa all'interpretazione dei messaggi, ed infine un'istanza della classe *LogFile*, il cui scopo verrà descritto in seguito.

A questo punto il bot può essere finalmente avviato, richiamando il metodo *startReceiving()* sull'oggetto *TelegramBotClient*. Dal momento che non si dispone di un indirizzo ip pubblico, la ricezione dei messaggi avverrà secondo la modalità "long polling", già illustrata nel capitolo 2. Fatto ciò, si stabilisce anche la connessione con Qlik Sense (con la versione

desktop), richiamando il metodo *QSCConnectDesktop()* sull'oggetto *QSApp* definito in precedenza. In concomitanza alla connessione a Qlik, viene anche aperta l'app di default, che nel nostro caso coincide con l'app "Flow".

Una volta terminate le operazioni di inizializzazione, il bot risulterà in attesa di eventuali messaggi provenienti dai client. Come già detto, gli input degli utenti possono sollevare due callback, le quali, pur reagendo ad eventi differenti, risultano molto simili ed è per questo motivo che nel seguito del presente paragrafo sarà presa in considerazione solo quella *BotOnMessageReceived*. Ogni volta che viene ricevuto un nuovo messaggio, viene dapprima valutata la tipologia del messaggio (solo i messaggi testuali vengono accettati), dopodiché vengono fatte delle differenziazioni sulla base del mittente: se il mittente, identificato per mezzo del "Telegram ID", risulta essere un amministratore (Telegram admin o bot admin), allora si confronta il messaggio ricevuto con i "comandi speciali", ossia una serie di comandi, non rilevanti, utili alla gestione del bot. Se, invece, l'identificativo del mittente non coincide con quello di uno dei due admin, allora la sua richiesta viene inoltrata direttamente al modulo *Conversation*, richiamando il metodo asincrono *Reply()* sull'oggetto *TelegramConversation*. Il metodo *Reply()* restituirà un oggetto di tipo *Response*, contenente la risposta corrispondente alla richiesta dell'utente. Questo risposta, però, non viene inviata al client, ma viene utilizzata come input per la funzione *ProcessConversationResponse()*, la quale consente di interpretare il contenuto dell'oggetto *Response* e di costruire un messaggio di risposta che sarà inviato al client, richiamando sull'oggetto *TelegramBotClient* uno dei metodi messi a disposizione dalla libreria *TelegramBot* (*SendTextMessageAsync()*, *SendPhotoAsync()*, *SendDocumentAsync()*, etc.).

Dal punto di vista del modulo Bot, dunque, il modulo *Conversation* è una "black box" cui vengono passati degli input (i messaggi da interpretare) e che fornisce, in output, le risposte che il modulo Bot dovrà andare ad inoltrare agli utenti.

4.2.2 Modulo *Conversation*

Il modulo *Conversation* riceve, dunque, attraverso la chiamata alla funzione *Reply()*, i messaggi degli utenti e deve tornare al chiamante un oggetto di tipo *Response*. Per poter "calcolare" la risposta ad una certa richiesta, però, il modulo *Conversation* ha bisogno di rivolgersi al modulo *NLP* ed è proprio per questo motivo che al suo interno viene creata un'istanza della classe *NLP*. Attraverso questo oggetto, infatti, è possibile richiamare le api messe a disposizione dalla libreria *Api.AiSDK*, le quali permettono al back-end di comunicare con un apposito agent, precedentemente configurato in *Dialogflow*, che svolgerà tutte le operazioni necessarie all'interpretazione dei comandi formulati in linguaggio naturale.

Per ciascun messaggio ricevuto, viene chiamata prima la funzione *UnderstandSentence()*, la quale internamente richiama, sull'oggetto *NLP*, il metodo *Predict()*. Questo metodo ritorna un valore booleano, che serve ad indicare se l'interpretazione della richiesta dell'utente è riuscita (true) o meno (false). In caso di successo, l'oggetto istanza della classe *NLP* risulterà "riempito" con i parametri necessari a calcolare la risposta, ossia un intent ed un insieme di valori aggiuntivi (misura, dimensione, elemento, etc.).

A questo punto ci sono due possibilità: la risposta da inoltrare al modulo Bot è già stata fornita dal modulo *NLP* e, in tal caso, non sono richieste elaborazioni aggiuntive, oppure la risposta va calcolata a partire dai parametri forniti dal modulo *NLP*, interrogando l'app *Qlik Sense*. Trattandosi di un bot aziendale, il cui scopo è proprio quello di interfacciarsi con *Qlik*, appare chiaro che lo scenario più ricorrente è il secondo ed è proprio per questo motivo che la maggior parte degli intent prevede che il modulo *Conversation* contatti il modulo *Qlik Sense* per accedere ai dati delle app.

Di seguito non vengono descritti tutti gli intent supportati dal modulo Conversation, ma solo quelli che risultano maggiormente esplicativi e che comportano un'interrogazione delle applicazioni Qlik Sense:

- KPI: questo intent viene restituito quando l'utente richiede il valore di una misura (ad esempio “dammi gli investimenti”) e, di conseguenza, all'interno dell'oggetto *NLP* sarà presente una misura. In questo caso, per prima cosa, si vanno a recuperare gli eventuali filtri, impostati precedentemente dall'utente, dopodiché vengono chiamati sull'oggetto *QSApp* i metodi *GetMasterMeasure()* e *GetExpressionFormattedValue()*, attraverso cui viene recuperato il valore corrispondente alla misura, vengono applicati gli eventuali filtri e, utilizzando il formato corretto, viene costruita la stringa da inoltrare al modulo Bot.
- Chart: questo intent viene restituito quando l'utente richiede di conoscere il valore di una specifica misura, declinata rispetto ad una dimensione. Come fatto anche per il caso precedente, per prima cosa vengono recuperati eventuali filtri impostati dall'utente, dopodiché, nel caso in cui siano state fornite dal modulo NLP sia la misura che la dimensione, vengono richiamati sull'oggetto *QSApp* i metodi *GetMasterDimension()*, *GetMasterMeasure()*, *QSSearchObjects()* e *GetDataList()*, i quali permettono di recuperare le informazioni relative all'intersezione tra la misura e la dimensione specificate dall'utente. Nel caso in cui mediante la funzione *QSSearchObjects()* fosse stata trovata, all'interno dell'app Qlik Sense, una visualizzazione avente misura e dimensione analoghe a quelle specificate dall'utente, la risposta costruita dal modulo Conversation comprenderebbe, oltre al valore puntuale dato dall'intersezione, anche il relativo grafico.
- Measure4Element: questo caso è simile al precedente, ma l'utente, anziché richiedere un valore di misura rispetto ad una dimensione, indica anche l'elemento, ossia il valore rispetto al quale fissare la dimensione (ad esempio “investimenti 2016”). Per il calcolo della risposta da inoltrare al modulo Bot, il modulo Conversation deve prima recuperare, mediante la chiamata al metodo *QSSearchInDimension()*, la dimensione cui applicare l'elemento specificato dal client, dopodiché, chiamando la funzione *AddFilter()*, va ad applicare il filtro e solo alla fine, può ottenere il valore di misura cercato, andando a richiamare i metodi già definiti per l'intent “KPI”. Può capitare che nella richiesta utente non si riesca ad individuare la dimensione o la misura. Nel primo caso vengono ricercate, all'interno dell'app, utilizzando la funzione *QSSearch()*, tutte le dimensioni cui risulti possibile applicare l'elemento fornito dall'utente; nel secondo caso, invece, viene utilizzata come misura l'ultima misura richiesta dall'utente.
- Filter: questo intent viene generato quando il messaggio dell'utente contiene una selezione (ad esempio “anno 2018”). Il modulo Conversation non deve far altro che richiamare sull'oggetto *QSApp* i metodi *QSSearchInDimension()* e *AddFilter()* per aggiungere il filtro indicato dall'utente. Può capitare che la richiesta dell'utente contenga soltanto l'elemento, senza specificare la dimensione; in questo caso, prima di applicare la selezione, dovrà essere effettuata una chiamata alla funzione *QSSearch()*, utile a ricercare una dimensione per cui abbia senso fissare il valore che l'utente ha fornito.
- ClearAllFilters: questo intent viene restituito quando l'utente richiede la cancellazione di tutte i filtri creati in precedenza. In questo caso l'interazione con Qlik Sense non è necessaria, in quanto i filtri sono memorizzati localmente e vengono applicati ogniqualvolta venga richiesto il valore di una misura.
- ClearDimensionFilter: questo intent viene restituito quando l'utente richiede la cancellazione dei soli filtri applicati rispetto ad una dimensione specifica. Per gli stessi motivi indicati prima, l'interazione con Qlik Sense è assente.

- CurrentSelections: questo intent viene restituito quando l'utente richiede di conoscere l'insieme di filtri che risultano applicati. In questo caso, tutto ciò che viene fatto dal modulo Conversation è costruire una stringa contenente le coppie dimensione – elemento, memorizzate localmente.

Una volta terminata l'interazione con il modulo Qlik Sense, il modulo Conversation costruisce, a partire dalle informazioni estratte dall'app, una risposta che sarà poi inoltrata al modulo Bot. La consegna della risposta al client non è prerogativa del modulo Conversation, sarà il modulo Bot a preoccuparsene, sfruttando le api messe a disposizione dalla libreria TelegramBot.

4.2.3 Modulo Qlik Sense

Come è possibile evincere dal nome, il modulo Qlik Sense espone un insieme di metodi che, internamente, richiamano le api messe a disposizione dalla libreria QlikSense.NetSDK. Le funzioni realizzate all'interno di questo modulo, dunque, consentono di stabilire una connessione con Qlik Sense (sia con la versione desktop che con la versione server) e di effettuare operazioni direttamente sulle app disponibili. Le operazioni di alto livello che sono state rese disponibili sono le seguenti:

- Creazione di una selezione: dati in input una dimensione ed un valore, viene applicata in Qlik Sense la selezione corrispondente.
- Connessione alla versione server con header authentication: si stabilisce una connessione con il Qlik Sense server, utilizzando un'autenticazione di tipo header authentication.
- Connessione alla versione server con credenziali NTLM: si stabilisce una connessione con il Qlik Sense server, utilizzando un'autenticazione del tipo NTLM.
- Connessione alla versione desktop: si stabilisce una connessione con la versione desktop di Qlik Sense, raggiungibile all'indirizzo "127.0.0.1:4848". In questo caso non viene realizzata nessuna procedura di autenticazione.
- Apertura applicazione: viene aperta una delle applicazioni disponibili all'interno dell'istanza di Qlik Sense cui si è connessi. I metodi definiti permettono sia di aprire un'app specifica, sfruttando l'id associato ad ogni applicazione, sia di aprire un'app di default, qualora non fosse specificato nessun identificativo.
- Creazione e rimozione di un foglio: viene fornita la possibilità di aggiungere nuovi fogli all'app aperta o, in alternativa, di cancellarli.
- Creazione di visualizations: vengono esposti una serie di metodi utili a creare nuove visualizzazioni all'interno dell'app aperta. Tra gli elementi grafici che è possibile creare figurano i KPI, le caselle di testo, i grafici a barre, i grafici lineari, i grafici a torta, le tree-map, le tabelle pivot, gli scatter chart e le caselle di filtro.
- Estrazione dei master items: viene data la possibilità di estrarre dall'applicazione aperta le master measures, le master dimensions e le master visualizations, memorizzandole all'interno di una struttura dati locale.
- Ricerca di un master item: viene data la possibilità di cercare per nome, all'interno dell'app, master measures, master dimensions e master visualizations.
- Ricerca di una dimensione in base all'elemento: viene definita una funzione che permette di trovare la dimensione corrispondente ad un certo valore di elemento. Se ad esempio il valore dell'elemento è "2018", allora la presente funzione tornerà la dimensione "Anno". Ovviamente potrebbe verificarsi il caso in cui più dimensioni presentano uno stesso valore

di elemento; in questa situazione, la funzione fornisce una lista di dimensioni, anziché un'unica dimensione.

- Creazione di una lista di dati: dati in input una misura, una dimensione ed un insieme di selezioni, viene fornito un metodo per ottenere da Qlik l'ipercubo costruito a partire da tali informazioni.

4.2.4 Modulo NLP

Questo modulo contiene un'istanza della classe *ApiAi* (definita all'interno della libreria *Api.AiSDK*) che consente di stabilire una connessione con l'agent definito in Dialogflow e di sottoporre a questo agent le richieste formulate dagli utenti in linguaggio naturale. L'agent ha a disposizione tutte le informazioni utili ad individuare l'intent corrispondente a ciascuna richiesta e ritorna al modulo NLP le entità eventualmente contenute nei messaggi (misure, dimensioni, elementi etc.). L'agent viene contattato mediante la chiamata alla funzione *TextRequest()*, la quale restituisce un oggetto di tipo *AIResponse*. Il contenuto dell'oggetto *AIResponse* viene poi estratto e sia l'intent che gli altri parametri vengono caricati all'interno dell'oggetto istanza della classe *NLP*, così da essere accessibili anche al modulo *Conversation*.

4.2.5 Modulo Log

Un'istanza della classe *LogFile* viene creato, all'avvio dell'applicativo, internamente al modulo Bot, dopodiché viene richiamato da tutti i moduli quando risulta necessario registrare una determinata operazione. Lo scopo del modulo Log, infatti, è quello di tenere traccia di tutte le attività svolte, dalla connessione di un utente alla ricezione di un determinato messaggio. Questa operazione risulta importantissima, specialmente nella fase iniziale, in quanto consente di monitorare il flusso di esecuzione seguito dal programma e permette di individuare eventuali errori o comportamenti anomali. Ad ogni nuovo avvio del programma, viene creato un nuovo file di log, memorizzato all'interno di una directory specifica, definita a priori.

4.3 Customizzazione del Chat Bot sulla base dell'app “Flow”

L'applicativo realizzato recupera alcuni dei parametri che ne consentono il funzionamento all'interno di un file di configurazione. Tale file è denominato “QlikSenseBot.exe.config” e deve essere memorizzato nella stessa cartella che ospita il file eseguibile. Tra i parametri che vanno specificati in questo file i più importanti sono:

- *cntBotToken*: è il token che identifica il bot creato sul server Telegram ed è necessario per lo scambio di messaggi tra il back-end ed i client Telegram.
- *cntApiAiKey*: è il token che permette di identificare l'agent corretto sulla piattaforma Dialogflow.
- *DemoqsAppId*: è l'id dell'app Qlik Sense che si vuole indicare come applicazione di default. Nel caso considerato, l'id inserito è quello corrispondente all'app “Flow”.
- *DemoqsAppName*: è il nome dell'app Qlik Sense che si vuole indicare come applicazione di default. Anche in questo caso, il nome inserito è quello dell'app “Flow”.
- *DemoqsServer*: è l'indirizzo del server che ospita l'istanza di Qlik Sense cui connettersi. Nel caso di connessione alla versione desktop non è necessario configurare questo parametro, in quanto l'indirizzo è impostato in modo statico all'interno del codice.

Oltre alla compilazione di questo file, un'altra operazione fondamentale ai fini del corretto funzionamento del bot riguarda la creazione dell'agent in Dialogflow. Progettare un agent nel modo corretto, infatti, può voler dire avere meno errori in fase di interpretazione dei messaggi degli utenti ed è per questo motivo che non bisogna sottovalutare questa fase. Relativamente allo sviluppo dell'agent da integrare con il Chat Bot, è stato necessario per prima cosa assegnargli un nome e, a tal proposito, la scelta è ricaduta su "QlikSenseBotRP", dove "RP" sta per "Rai Pubblicità". Successivamente si è passati alla definizione delle developer entities che, come già spiegato all'interno del capitolo 2, sono assimilabili a delle variabili e vengono riempite con il contenuto di appositi valori estrapolati, di volta in volta, dai messaggi forniti dagli utenti. Per l'agent "QlikSenseBotRP" sono state create le quattro entities indicate di seguito:

- @ChartType: questa entity viene utilizzata ogniqualvolta l'utente, all'interno dei propri messaggi, fa riferimento ad una specifica tipologia di visualization. Tra i valori campione specificati in fase di definizione della entity si è cercato di inserire "PivotChart", "BarChart", "LineChart", "PieChart", etc.
- @Dimension: come per la entity precedente, anche questa viene utilizzata quando l'utente inserisce, nei messaggi inviati al bot, uno o più valori di dimensione. A differenza del caso precedente, però, per adattare l'agent al funzionamento con l'app "Flow", è stato necessario specificare, come valori campione per questa entity, tutte le master dimension che sono state definite all'interno dell'applicazione Qlik. Si è pensato, poi, di definire anche dei sinonimi per ciascuna dimensione, in quanto esiste la possibilità concreta che l'utente si riferisca ad una dimensione utilizzando un nome diverso da quello utilizzato in Qlik.
- @Element: mentre la entity "dimension" ha un insieme finito di valori che l'utente può specificare (anche considerando i sinonimi), per quanto riguarda la entity "element", invece, non è stato possibile specificare tutti i possibili valori campione, data l'elevata cardinalità di questi ultimi.
- @Measure: la entity "measure" serve ad estrarre i valori di misura specificati dall'utente all'interno dei messaggi e, analogamente a quanto fatto per la definizione della entity "dimension", è stato necessario definire come valori campione, le master measures definite all'interno dell'app "Flow". In questo caso la configurazione è risultata semplice, in quanto l'app "Flow" presenta un unico valore di misura, ossia "Investimenti".

Dimension	Synonyms
Tipo periodo Nielsen	Tipo periodo, periodo
Concessionario	Concessionario, Concessionari
Semestre	Semestre
Tipo periodo Nielsen	Tipo periodo Nielsen
Settore Nielsen	Settore Nielsen, Settore, Settori, Settori Nielsen
Holding	Holding
Cliente Nielsen	Cliente Nielsen, Clienti Nielsen, Clienti, Cliente

Figura 21 - Creazione della entity "@Dimension"

Terminata la fase di creazione delle entities, è stato necessario definire anche gli intents, ossia i gruppi all'interno dei quali l'agent dovrà far ricadere i singoli messaggi ricevuti dagli utenti. In questa fase si è cercato di ipotizzare tutte le possibili richieste che gli utenti avrebbero potuto sottoporre al bot, definendo oltre agli intent relativi al mondo Qlik Sense, anche una serie di intent utili a portare avanti una conversazione informale. Per ciascun intent, poi, è stato necessario definire un insieme di training phrases, utilizzati dall'agent per affinare le proprie capacità di interpretazione.

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST	PROMPTS
<input checked="" type="checkbox"/>	Measure	@Measure	\$Measure	<input type="checkbox"/>	Define prompts...
<input checked="" type="checkbox"/>	Element	@Element	\$Element	<input checked="" type="checkbox"/>	Define prompts...
<input type="checkbox"/>	Dimension	@Dimension	\$Dimension	<input type="checkbox"/>	–
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>	–

Figura 22 - Creazione dell'intent "qs-ShowMeasureForElement"

Di seguito sono indicati alcuni degli intent realizzati per il QlikSenseBotRP:

- qs-ShowMeasure: è l'intent sotto il quale ricadono i messaggi in cui l'utente richiede un valore di misura. L'unica entity che è possibile nel corpo del messaggio è "@Measure" e l'azione che viene sollevata una volta che l'intent è stato individuato prende il nome di "ShowMeasure". Alcune delle training phrases inserite per questo intent sono "Qual è il valore degli investimenti?", "Mostrami gli investimenti", "Mostra gli investimenti", "Potrei avere gli investimenti".
- qs-ShowChart: questo intent è quello scelto quando l'utente richiede un'intersezione tra misura e dimensione, o quando specifica, oltre alla misura e alla dimensione, anche l'elemento. All'interno di questa tipologia di messaggio è possibile identificare tutte e quattro le entity definite per l'agent e l'azione che viene sollevata prende il nome di "ShowChart". Alcune delle training phrases inserite sono "investimenti per settimana", "investimenti per mezzo", "investimenti rai", "dammi gli investimenti per mese del cliente Ford".

- qs-ShowMeasureForElement: questo intent viene scelto quando l’utente richiede l’intersezione tra una misura ed una dimensione, specificando anche il valore che la dimensione deve assumere (l’elemento). All’interno di questo tipo di messaggi è possibile identificare le entity “@Measure”, “@Dimension” ed “@Element”. L’azione sollevata è la “ShowMeasureForElement” e alcune delle training phrases aggiunte sono “quali sono gli investimenti su Rai uno di questo mese”, “vorrei gli investimenti del 2015”, “investimenti holding Barilla”, “investimenti mese di marzo”.
- qs-Filter: questo intent è selezionato quando l’utente invia al bot un messaggio di selezione, ossia quando specifica una dimensione ed il valore da assegnare a tale dimensione. Le entity che si possono estrarre sono “@Dimension” e “@Element” e l’azione sollevata è la “Filter”. Alcune training phrases utilizzate sono “Anno 2016”, “Mese luglio”, “ottobre”, “cliente Bolton”.
- qs-ClearAllFilters: questo intent viene scelto quando l’utente richiede di eliminare tutte le selezioni applicate. In questo caso non ci sono entity che è possibile estrarre e le training phrases utilizzate sono “rimuovi filtri”, “rimuovi tutti i filtri”, “elimina selezioni”.
- qs-ClearDimensionFilter: questo intent è simile al precedente, ma si differenzia in quanto l’utente, oltre a dichiarare che vuole rimuovere un filtro, indica anche la relativa dimensione. In questo caso l’unica entity identificabile all’interno dei messaggi è la “@Dimension” e alcune delle training phrases utilizzate sono “rimuovi mese”, “rimuovi anno”, “togli filtro su anno”.

qs-CurrentSelections: questo intent viene sollevato ogniqualvolta l’utente richiede di conoscere le selezioni attive. Anche in questo caso non ci sono entity che possono essere ricercate all’interno dei messaggi e alcune delle training phrases utilizzate sono “dammi la selezione corrente”, “selezione corrente”, “cosa è selezionato”, “che filtri ci sono”.

Agli intent dedicati al chat bot sono stati poi aggiunti i cosiddetti intent di “small talk”, ossia degli intent generici che possono offrire supporto a richieste utente non previste in fase di sviluppo.

Conclusioni

Il lavoro descritto nel presente elaborato di tesi mi ha consentito di migliorare sia da un punto di vista tecnico che da un punto di vista umano, permettendomi di comprendere le dinamiche che sono alla base del lavoro aziendale e della cooperazione. Il confronto con i colleghi più esperti, all'interno di un ambiente serio e stimolante, mi ha permesso, inoltre, di acquisire tantissime nuove conoscenze, senza precludermi la possibilità di apportare il mio contributo personale, laddove possibile.

La prima fase del lavoro, volta allo studio del sistema di Business Intelligence in Rai Pubblicità e allo sviluppo delle app Qlik Sense “AQC” e “Flow” mi ha dato la possibilità di seguire in prima persona tutti gli step previsti dall'iter produttivo aziendale, dall'analisi delle specifiche alla consegna del prodotto finito, responsabilizzandomi e insegnandomi l'importanza della pianificazione delle operazioni e del rispetto delle tempistiche. La parte relativa allo sviluppo del Chat Bot, invece, ha fatto sì che fossi libero di mettere le conoscenze, da me acquisite in abito universitario, a disposizione dell'azienda, introducendo, all'interno del già vasto arsenale di tecnologie aziendali, uno strumento che risultasse utile e, allo stesso tempo, innovativo.

Per quanto riguarda i possibili sviluppi futuri, data la vastità dell'app “AQC”, si potrebbe decidere di trasformare ciascun foglio un'applicazione a sé stante, ripensando l'app stessa come un “hub” di carattere generale da cui partire per “saltare” alle applicazioni esterne, volte all'analisi di ambiti di investimento specifici (un po' come è stato fatto con l'app “Flow”). Tra le altre cose, questa soluzione porterebbe ad una semplificazione del data model attuale, il quale verrebbe alleggerito e le informazioni sarebbero ridistribuite tra i modelli dati delle singole app.

Relativamente al bot, invece, un possibile miglioramento potrebbe essere costituito dal trasferimento del software sviluppato su uno dei server aziendali, andando a sostituire il collegamento alla versione desktop di Qlik Sense, con un collegamento alla versione server; in questo modo, sarebbe garantito l'accesso a tutte le applicazioni sviluppate dal team di BI, anziché alla sola app “Flow”. Fatto ciò, si potrebbe anche pensare di effettuare una fase di beta testing, consentendo l'uso del bot ad un sottoinsieme limitato di utenti, così da identificare eventuali bug e da andare, contemporaneamente, ad arricchire gli insiemi di training phrases, utili ad affinare la capacità dell'agent di Dialogflow di interpretare le richieste formulate in linguaggio naturale. Un vero e proprio passo in avanti, infine, potrebbe essere rappresentato dalla trasformazione del bot, da semplice esperimento, a strumento effettivamente utilizzato dall'utenza finale, per la fruizione delle informazioni. Tale decisione potrebbe costituire un vero e proprio punto di svolta, non solo per il team di BI, ma per l'intera azienda, in quanto si andrebbe ad investire su un prodotto nuovo, giocando d'anticipo sui principali competitors.

Bibliografia e sitografia

Rezzani A., *Business Intelligence*, Apogeo, 2012

Howson C., *Successful Business Intelligence. Unlock the Value of BI & Big Data*, McGraw-Hill, Milano, 2013

Codice dei contratti pubblici, d.lgs. n.50/2016

Massari A., Montalti M., Oliveri A. P., *L'accordo quadro negli appalti pubblici. Analisi teorico-normativa e modelli operativi*, a cura di Massari A., Maggioli Editore, Rimini, 2013

Manenti E., *La Business Intelligence in Rai Pubblicità - Studio del framework di ETL e realizzazione del data model Qlik nell'ambito di un caso reale*, 2018

Rai Pubblicità, <<http://www.raipubblicita.it/azienda/>>, ultima consultazione: 09/06/2018

Wikipedia: Italia, <https://it.wikipedia.org/wiki/Business_intelligence>, ultima consultazione: 10/06/2018

Data Skills, <<https://www.dataskills.it/business-intelligence/cos-e-business-intelligence/>>, ultima consultazione: 10/06/2018

My Bi, <<http://www.mybi.it/BusinessIntelligence/tabid/54/language/it-IT/Default.aspx>>, ultima consultazione: 10/06/2018

B.I. Strategy, <http://www.bistrategy.it/app_bi_origini_definizione.aspx>, ultima consultazione: 10/06/2018

Computer world, <<https://www.cwi.it/applicazioni-enterprise/business-intelligence-bi/business-intelligence-definizione-101139>>, ultima consultazione: 10/06/2018

Smart Data Collective, <<http://www.smartdatacollective.com/here-s-why-natural-language-processing-future-bi/>>, ultima consultazione: 12/06/2018

Code burst, <<https://codeburst.io/2-how-assistant-work-introduction-to-dialogflow-319a72ba2db>>, ultima consultazione: 12/06/2018

DataBase and Data Mining Group: Politecnico di Torino, <<http://dbdmg.polito.it/wordpress/teaching/>>, ultima consultazione: 12/06/2018

Oracle, <<https://www.oracle.com/database/technologies/appdev/sql-developer.html>>, ultima consultazione: 12/06/2018

Qlik: Help, <<https://help.qlik.com/it-IT/sense/February2018/Content/Home.htm>>, ultima consultazione: 28/06/2018

Dialogflow: Documentation, <<https://dialogflow.com/docs/getting-started>>, ultima consultazione: 13/06/2018

Telegram: Api Documentation, <<https://core.telegram.org/>>, ultima consultazione: 13/06/2018

Wikipedia: England, <<https://en.wikipedia.org/wiki/Dialogflow>>, ultima consultazione 12/06/2018

Wikipedia: Italia, <https://it.wikipedia.org/wiki/Elaborazione_del_linguaggio_naturale>, ultima consultazione: 12/06/2018

Ontotext, <<https://ontotext.com/knowledgehub/fundamentals/dikw-pyramid/>>, ultima consultazione: 10/06/2018

Qlik Branch, <<https://branch-blog.qlik.com/all-you-need-to-know-about-qlikbotnet-b86b455f0e10>>, ultima consultazione: 26/06/2018

Wikipedia: England, <https://en.wikipedia.org/wiki/Data_visualization>, ultima consultazione: 28/06/2018

Github inc., <<https://github.com/stefanwalther/sense-navigation>>, ultima consultazione: 26/06/2018

Ringraziamenti

Questo lavoro di tesi non è soltanto il risultato di alcuni mesi di duro lavoro, costituisce bensì il punto di arrivo di un lungo percorso di crescita e maturazione, sia umana che professionale, che mi ha visto protagonista e che non sarebbe stato possibile senza l'aiuto di alcune persone che sento la necessità di ringraziare.

Prima di tutto ringrazio la professoressa Elena Baralis, la quale si è dimostrata sempre cortese e disponibile nei miei confronti, sia in veste di docente, durante il corso di studi, sia in qualità di relatrice, durante il lavoro di tesi.

Un grande ringraziamento va poi a Mauro Sandri, che è stato costantemente presente durante la fase di realizzazione della tesi, correggendomi laddove sbagliavo e guidandomi nelle situazioni più difficili.

Ringrazio anche tutti i componenti del gruppo BI di Rai Pubblicità e dell'ufficio IT, i quali si sono sempre dimostrati più che disponibili ad aiutarmi e a darmi consigli preziosi.

Un immenso riconoscimento lo devo alla mia famiglia, in particolare ai miei genitori e a mia sorella Lia. Li ringrazio per aver sempre creduto in me e sono certo che non avrei mai potuto raggiungere questo traguardo senza il loro appoggio e, soprattutto, senza il loro affetto.

Ringrazio Alessandra, il mio pensiero felice nei momenti di difficoltà, la mia roccia e ancora di salvezza. Nel mio essere costantemente indeciso, l'unica certezza, il punto fermo, è lei e lo sarà sempre.

Un ringraziamento particolare va al mio amico e collega Enrico, che nonostante il mio essere ritardatario, pigro e talvolta irritante, mi ha accompagnato sia nel percorso universitario che in quello lavorativo.

Ringrazio Alessio F., Alessio C., Tano, Dario e Melo, che si sono dimostrati dei veri amici, oltre che semplici colleghi universitari e coinquilini, e che sono diventati a tutti gli effetti la mia famiglia a Torino.

Ringrazio infine gli amici di sempre che, proprio grazie alla distanza che ci separa, mi hanno fatto capire che l'amicizia è qualcosa che trascende tutto il resto.